METHOD AND SYSTEM FOR ENHANCED PACKET TRANSMISSION IN CELLULAR NETWORKS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application is related to copending commonly assigned U. S. Patent Application Serial No. ______________, which was filed on November 1, 2001, and entitled "Method And System For UMTS Packet Transmission Scheduling On Shared Downlink Channels".

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to communications; in particularly, packet transmission scheduling for cellular networks and to a packet transmission scheduling system having such efficient packet transmission scheduling functionality.

2. Description of the Prior Art

There are problems associated with mobility handling in packet switched networks having scheduling systems, especially in packet switched cellular mobile communication systems, and particularly Code Division Multiple Access systems (CDMA) like Universal Mobile Telecommunication Systems (UMTS) or Time Division Multiple Access systems (TDMA) like General Packet Radio Systems (GPRS).

As known, in packet switching networks the task of multiplexing essentially reduces to the task of ordering packets and to then send them serially over a shared link. This process of serialization is referred to as scheduling. For certain network links, especially for wireless links a certain amount of pre-given link characteristics have to be applied to protocol data units (PDU) to be transmitted, that is traditionally addressed by a segmentation of layer-3-PDU performed by a layer-3 scheduler, whereas a lower so called MAC-scheduler (medium access control-scheduler) provides the medium access control including the provision of respective transport blocks for a transmission data flow via the Physical-layer (PHY-layer).

However, recent scheduling methods consist of independent scheduling systems of each cell within a radio network controller (RNC), such as of an UMTS based network for example, and accordingly, there is in general no interconnection between these schedulers. In case of a handover procedure for example, i.e. when a mobile station is moving out of the coverage of a current cell and hence, has to be handed over towards a new cell which is better able to serve the data flow of the mobile station, within such a recent data transmission scheduling scheme the handover procedure may be described as follows:

Since there is no interconnection between the scheduling systems of different cells, the handover procedure has to be performed by means of a central instance. In packet switched networks this is usually the Service Gateway Sub Node (SGSN).

During the handover procedure the SGSN sends the layer-3 PDUs towards the scheduler of the new or target cell starting with the last layer-3 PDU, which was not completely transmitted by the old cell scheduler.

Then, the target cell scheduler starts to transmit beginning with the first transport block that is segmented from the layer-3 PDU.

By receiving of the transport blocks from this PDU the mobile station has to discard the transport blocks from the PDU that was not entirely transmitted by the scheduler of the old cell.

Since the handover process is performed over a central instance, there is a significant delay for establishing the data transmission in the new cell resulting in an non-continuous data flow with regard to the mobile station and hence for its user. Moreover since the new cell scheduler is starting its transmission always with the first transport blocks from the current layer-3 PDU there is a waste of resources, too.

Since the scheduling status is not transferred to the scheduler serving the new cell, the fact that the flow was potentially backlogged, i.e. it has received less service in the past than it has requested, is not taken into account by the new scheduler.

SUMMARY OF THE INVENTION

The present invention provides an improved approach for packet transmission scheduling, avoiding the above mentioned problems by simultaneously optimizing the

transmission of data flows within packet switched cellular systems, in particular adapted to be used for UMTS and/or GPRS systems.

A QoS scheduling is used for handling multiple data flows in packet switched cellular systems, especially in a packet switched mobile telecommunication system, wherein the scheduling mechanism of a specific cell is coupled with the scheduling mechanism of at least one second cell, in particular by transferring data between the specific cell and at least one second cell comprising a status information concerning an actual data flow within the specific cell.

By the coupling of scheduling mechanisms for different cells the use of information from the past of a data flow for an actual scheduling process is provided. Consequently, by an additional use of the inventive status transfer a continuous data flow of higher layer PDUs is achieved. Furthermore, in particular in case of performing a handover of a mobile station, a scheduling means of a target cell is provided, that knows about the current status of served data flows and hence, the transport blocks that were already successfully received by the mobile station must not be newly transmitted resulting in a further optimization of usable radio resources.

By coupling the scheduling mechanism of the target cell, the information about the current status of the served data flow can be used for optimizing the data transmission of that flow within the target cell that may additionally result in an inter-cell compensation of a data flow rate assigned to a user.

Accordingly, by the compensation of data flow rates a required quality of services (QoS) of the data flow is guarantied while optimizing the resource usage. Moreover, it is ensured, that all data flows are served having the requested QoS. If there is still capacity free then the scheduling means handles a compensation or maximal data flow rate. On the other hand side, the same principle can also be used for lowering a data flow rate a certain time for mobile users that came from a low loaded cell into a higher loaded cell in order to give more resources to the other users that are suffering from high loading.

According to one embodiment, the invention provides two scheduling mechanisms per cell, which are linked together such, that a scheduling means of an upper layer provides a certain degree of predictable behavior and a scheduling means of a lower layer provides Medium Access Control (MAC access) and also allows for bandwidth conserving segmentation and allocation strategies.

Advantageously, dependent on the specific network or system these scheduling means may be all part of terrestrial portions of the radio network and may be located within one network element or in separate elements. Therefore, the scheduling approach is adapted to allow a status transfer within scheduling means of one network element and/or a transfer of the status information between different elements, for example, even an inter-RNC handover is supported which is seen as critically in a UMTS system.

This technique may be used in networks having a single upper layer scheduling means per cell cluster and/or a separate upper layer scheduling means per cell.

In particular in UMTS-based networks, with one network controller handling the data flow of a large number of cells it is proposed to provide a single upper layered scheduling means per cell cluster. If however the lower layered scheduling means is located in different network elements causing a difficult handling of the data flows within one single upper layered scheduling means, it is suggested to provide a separate upper layered scheduling means per cell. Moreover, a further advantage thereof is, that a handling of timestamp drifting between the cells might show improved performance, since all search and sort operations are done on a smaller set of items.

In yet another embodiment, a combination of a coupled Layer 3 and MAC (Medium Access Control) layer is used. For the basics of such a combination reference is made to the co-pending European Patent Application 00 310 344.7, "Method of linking two schedulers of a multi-layer network and a network comprising a transceiver having linking functionality for two schedulers" and to the co-pending European Patent Application 00 310 343.9, "Method and System for UMTS Packet Transmission Scheduling on shared Downlink Channels".

Thus, a further enhanced scheduling method is provided which is especially adapted to CDMA-based or TDMA-based mobile communication systems, using a significantly improved adaptation of the basic scheduling method. The contents of European Patent Applications 00 310 343.9 and 00 310 344.7 are incorporated in the disclosure of the present application as Appendix A and B, respectively.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is described in more detail below and reference is made to the accompanying drawings, in which

Fig. 1 shows an exemplary signaling flow for a status information update of a UMTS-based network,

Fig. 2 shows an example concerning the principle of an inter-cell compensation,

Fig. 3 shows a first preferred approach according to the invention using one PDU-scheduler per cell cluster,

Fig. 4 shows an exemplar synchronization of data flows between cells by timestamp shifting, and

Fig. 5 shows a second preferred approach according to the invention using a separate PDU-scheduler per cell.

DETAILED DESCRIPTION OF THE INVENTION

Referring next to Fig. 1 a main idea of coupling schedulers over cell boundaries for improving the mobility handling is shown. According to the example of Fig. 1, the coupling is achieved by an explicit transfer of status information about the data flow to be handed over from a current scheduler S_c towards the scheduler S_t of a new or target cell.

It has to be noted, that only the control signaling message data flows are depicted. For simplicity reasons, any eventually necessary acknowledgement (ACK) and/or negative acknowledgement (NACK) forming an important part of a signaling message transfer are neglected, since they are generally known by a person skilled in the art. In the following exemplar description, however, it is always assumed that acknowledgements (ACKs) and/or negative acknowledgements (NACKs) are properly handled by the system.

The example of an information update during a handover procedure, as shown in Fig. 1, incorporates one mobile station MS and one radio network subsystem (RNS) of an UMTS-based network. Within the radio network subsystem there is a radio resource management entity RRM, which has the control of the radio resources

of a certain number of cells. Usually, the handover decision functionality is incorporated within the radio resource management block RRM.

The first scheduler S_c belongs to the current cell which serves the data flows of the mobile station MS prior to a handover, whereas the second scheduler S_t handles the data flows for the mobile station of a new or target cell. According to the example, the radio resource management RRM, the first and second scheduler S_c or S_t are all part of the terrestrial radio network. They can be located within one network element, e.g. one radio network controller(RNC) or in separate elements, e.g. two or more radio network controllers (RNCs).

The basic signaling message data flow preferably consists of the following steps relating to the reference signs of Fig. 1:

1. The mobile station MS sends certain measurement reports to the network. These reports normally contain information on the current quality of the radio link. Regarding CDMA systems like UMTS a reporting of the E_c/I_o ratio from the pilot channel of the cell can be used, wherein E_c denotes the received signal energy within the actual cell and I_o denotes the interference at the mobile station MS. Regularly reporting as well as reporting on events that are specified by the network are supported. For example, the measurement reports may be signaled to the resource management entity RRM by piggybacking data on currently scheduled data flows (in-band signaling) or via a separate traffic channel (out-of-band signaling).

2. Based on the reported measurement results the radio resource management entity RRM decides on the necessity to handover (HO) from the current cell towards another or target cell. For this handover decision several criteria can be used. For example, in CDMA systems like UMTS a comparison of reported signal quality in terms of the E_c/I_o ratio from the pilot channels of the different cells is widely used.

3. When the radio resource management entity RRM decides that a handover from the current cell towards a target cell has to be performed it creates a handover command message that will be sent to all involved entities, i.e. the mobile station MS, the scheduler S_c of the current cell and the scheduler S_t of the target cell. This message contains the information that a handover is necessary and the data which are necessary for the (re-)configuration of the entities, such as for example concerning the transport format set, the spreading factor etc., for example, as described in Appendix A. However, as mentioned above, in Fig. 1 only an example of the sequence of

messages is shown independent of the exact order since other solutions might also be used.

4. When the scheduler S_c of the current cell receives the handover command message it stops the scheduling procedure for the data flows of the mobile MS by removing its context from the scheduler S_c.

5. Then, the scheduler S_c of the current cell transfers the status information to the scheduler S_t of the target cell. Practically, the status information is including at least the identifiers of the data flow, the number of the layer-3 PDU and the number of the transport blocks (TBs) that were successfully acknowledged by the mobile station MS. Even if the Fig. 1 shows only the logical data flow, physically the flow may also pass the radio resource management entity RRM.

6. After the scheduler S_t of the target cell has received the status transfer message it starts the scheduling process also for the flows of the mobile station MS that were identified with the flow identifiers by including the context into the scheduler S_t. The starting transport block is the subsequent transport block with regard to the transport block last successfully acknowledged by the mobile station MS.

The message signaling flow description aforementioned does not consider the problem of rerouting the layer-3 PDUs from scheduler S_c towards scheduler S_t. However, from the scheduling point of view it has to be ensured that at the begin of step 6 the scheduler S_t has access to the layer-3 PDUs to be scheduled next for the mobile station MS. When using the inventive scheduling method with two coupled schedulers as described, for example, in Appendix A, the routing method depends on the kind of implementation of the PDU-scheduler. The inventive scheduling approach supports one single PDU-scheduler handling the data flows from a cluster of a certain number of cells as well as separate PDU-schedulers handling the data flows for one cell, only, as it is described in more detail below.

By the coupling of schedulers S_c and S_t, each of which associated to a different cell according to the invention it becomes possible to make use of the information from the past of a data flow for the actual scheduling process. This can be used in addition for the compensation of data flow rates in order to meet the required quality of services (QoS) of the data flow while optimizing the resource usage. Such methodology may be referred-to as "inter-cell compensation". The principle of inter-cell compensation is depicted in Fig. 2, showing a data flow rate R_n used for user #n

versus the time t. The drafted example is based on the situation, when a mobile user moves from a low loaded cell, represented by the area marked with I, to a higher loaded cell, represented by the area marked with II and then back to a low loaded cell, represented by the area marked with III. It is assumed that the used data flow rate R_n assigned to user #n in area I is equal to the data flow rate R_{QoS} according to the required QoS of the associated service. If the mobile station moves into the higher loaded area II, then according to the rate conserving policy, as described in Appendix A, the scheduler of this area assigns a data rate R_n which may be lower than the originally assigned one, i.e. lower than R_{QoS}. When the mobile user moves again to a lower loaded area III a normal scheduling method would assign the data flow rate R_{QoS} which is chosen according to the required QoS of the service. This is represented by the line referenced by 10a. In contrast to this conventional method, by use of the inventive approach, a higher data rate R_n represented by the line referenced by 10b, is assigned to the user in area III for a certain amount of time in order to allow him to "compensate" for the lower data rate he experienced in area II. This helps to still maintain the long term QoS of the associated service even in case the short term QoS is violated for a certain time period t_i.

The assignment of the data rates R_n and time duration of the "compensation" phase has to be done according to the decreasing of the flow rate R_n in area II and the time interval t_i. However, it has to be noted, that the duration of this time interval t_i is critical to the performance of the inventive method. Depending on the respective specific system and environment parameter, if the time interval t_i is too large, then the inter-cell compensation 10b may have no effect on the service QoS and needs therefore not to be used in this case.

Therefore, for the purpose of inter-cell compensation it is proposed to enhance the assignment of the transport format set (TFS) for the data flows and the scheduling policy with regard to the proposal of minimum and maximum values, as given, for example, in Appendix A.

According to the invention, three main values of a transport format are preferred, that are defined as follows:

- A minimum value, according to which the transport format has to be assigned regarding the minimum requirements to achieve just the requested QoS for each data service, as described for example in Appendix A.

- A compensation value, according to which a transport format may be assigned for the inter-cell compensation of data flows coming e.g. from highly loaded cells.

- A maximum value for a transport format to be used for optimizing the scheduling decision, when there are resources still available, e.g. for pro-active scheduling, as described for example in Appendix A.

As discussed in Appendix A, more transport formats than these principal ones might be assigned for higher granularity of data rates in order to reduce extensive padding. However, the MAC scheduling policy proposed in Appendix A, may be modified for doing inter-cell compensation: Therefore it is suggested, that the MAC scheduler takes the minimum transport formats with highest priorities, i.e. it tries to ensure that all data flows are served having the requested QoS. If there is still capacity free then the scheduler handles the "compensation traffic" and the "maximum traffic", where priority might be given to "compensation traffic".

The same principle can also be used for lowering a data flow rate a certain time for mobile users that came from a low loaded cell into a higher loaded cell in order to give more resources to the other users that are suffering from high loading.

A first very preferred implementation of the inventive scheduling approach comprises the implementation of a single PDU-Scheduler per cell cluster.

As known for a person skilled in the art, in a UMTS-based network, one radio network controller (RNC) handles the data flows for a large number of cells. Here, it might be reasonable to apply one PDU-scheduler not only for one cell, but also for a cell cluster that incorporates cells from a certain connected area. The principle of such a common scheduler is depicted in Fig. 3, according to which one PDU-scheduler is used for a cell cluster with an exemplar number of three cells.

According to the principles described in Appendix A, the upper PDU-scheduler operates on the input data from layer-3, the Protocol Data Units (PDU). It receives the QoS requirements of each data flow. When the availability of schedulable PDUs are notified to the QoS-scheduler it determines the order in which PDUs should receive service. Each of the MAC-scheduler serves only the PDUs that are related to its cell from this list and tries to reflect the order in the list, while also taking timing and power constraints into account. The MAC-scheduler is active at every frame, e.g. on a 10ms base, and schedules the data flows related to its cell. The PDU-scheduler is

operated on all active data flows of the cell cluster, i.e. with a non-empty PDU-flow-queue.

Since prior systems of serially uncoupled schedulers can show undesirable behavior, both schedulers are linked together by means that the MAC scheduling is driven by the PDU scheduler's state. This is shown as clouds in Fig. 3 for each MAC scheduler. Such linking causes that only one PDU list per cell cluster has to be maintained, that in case of performing a handover to a cell, which is also served by the same cluster, the scheduler S_t of the target cell (Fig. 1) can directly start with its transmission after it has received the status information transfer message from the scheduler of the old or former cell, because it accesses the same PDU list. The "routing" is simply done by the aforementioned status information transfer. Hence, an extra PDU rerouting is not necessary.

It is known, that a working with flow time stamps, as argued for example by J. Cobb, M. Gouda and A-EL-Nahas in "Flow timestamps", Annual Joint Conference of Information Sciences 1995 (Appendix C) eases the estimation of processing time since the upper limit of elements in the service-list of a PDU-scheduler can be limited to the maximum number of flows in the system. However, especially when using relative dynamic priorities for the PDU scheduling, as described for example in this identified document, the method of one single PDU scheduler may show undesirable behavior when doing handover as described in the following:

In some circumstances it happens that the priorities, which have the nature of a virtual timestamp in this example, drift away from each other for PDUs associated with different cells. Usually the timestamps are naturally kept in a close range, because always the leading packets are served, thus the distance of all timestamps is narrowed by the scheduling process. Due to the fact that only a PDU can be served by its recently serving cell this can lead to clustering within the serving list. One reason might be that within the cluster cells with low load and other cells with higher load are combined for PDU scheduling.

In this case the timestamps that are currently handled by a scheduler of a highly loaded cell may become much different from those handled by a scheduler of a low loaded cell. In terms of virtual clock scheduling the perceived service time in the highly loaded cell progresses slower than in the less loaded cell. Thus, the highly

loaded cell will result in PDUs with small timestamps and such high priorities, while the others were served in the past have large timestamps and such lower priorities.

Then, in case of handing over a user from e.g. a high loaded cell towards a lower loaded cell the scheduler would only grant service to the handover flow and fully compensate for its lack of service in the past. This may be only partially desirable. Although the radio resource allocation entity (RRA), as described in Appendix A, should try to avoid this by properly assignment of radio resources according to the current cell load and QoS requirements of the data flows, this situation still might occur. In this case a "synchronization" of the data flows between the cells when a handing over is performed is preferably applied. This synchronization procedure is done practically by shifting the time stamps when a handover is performed, as shown in Fig. 4. In Fig. 4, t_b is representing the time with regard to the old cell, i.e. before the handover is performed, and t_a is representing the time with regard to the new or target cell, i.e. after the handover is performed.

Accordingly, for maintaining a limited fair scheduling the difference between the timestamps of the current and target cell are decreased by timestamp shifting, wherein Fig. 4 shows the situation before and after the timestamp shifting. For achieving this defined goal of maintaining a limited fair scheduling, preferably a timestamp window is defined according to the timestamps of the currently served data flows. Regarding Fig. 4, the timestamp window is defined by

"min" identifying the minimum timestamp value of all PDUs of the currently scheduled data flows associated with this cell,

"max" identifying the maximum timestamp value of all PDUs of the currently scheduled data flows associated with this cell,

"δ" identifying a timestamp offset value that is used for inter-cell compensation as described above.

For performing the timestamp shifting the following steps are proposed:

If the timestamp of the data flow in the old cell is below the value determined by "min-δ" of the target cell, the timestamp will be shifted towards that "min-δ" when executing the handover towards the target cell;

If the timestamp of the data flow in the old cell is above the value determined by "max+δ" of the target cell, the timestamp will be shifted towards that "max+δ" when executing the handover towards the target cell;

If the timestamp of the data flow in the old cell is between the values respectively determined by "min-δ" and "max+δ" of the target cell, the timestamp will not be shifted when executing the handover towards the target cell.

By the usage of such an algorithm it is possible to limit the timestamp variations when handing over the mobile station from a current cell to another or new cell. However, by performing the shifting procedure a search operation over the service list in the PDU scheduler might become necessary.

When using the above described single PDU scheduler refinement the traffic to be compensated gets the highest priority and always will be served first from the MAC scheduler in the target cell.

Since, however, in some cases this might violate the desired scheduling policy. Thus, a very preferred refinement of the invention incorporates a separate PDU-Scheduler per cell, the principles thereof are illustrated by Fig. 5.

According to the above discussed it might be desirable to have a separate PDU scheduler per cell, in particular since

- the handling of the timestamp drifting between the cells might show improved performance with separate PDU schedulers, where all search and sort operations are done on a smaller set of items, and

- the handling of the data flows within one single PDU-scheduler is difficult, when the MAC schedulers are located in different network elements, especially caused by neighbor cells that are controlled by different radio network controllers (RNCs) when an inter-RNC handover is required.

Based on Fig. 5, showing the example of a separate PDU-Scheduler per cell of three cells, with an intra-RNC handover from one cell to another cell, as indicated by the arrow 100 and according to the principle described in Appendix A, each of the upper PDU-schedulers operates on the input data from layer-3, the so-called Protocol Data Units (PDU). Each of the PDU-Schedulers receives the QoS requirements of each data flow of the respective cell. When the availability of schedulable PDUs are notified to the QoS-scheduler it determines the order in which PDUs should receive service. The MAC-scheduler serves the PDUs from this list and tries to reflect the order in the list, while also taking timing and power constraints into account.

The MAC-scheduler is active at every frame, e.g. on a 10ms base. It schedules the data flows related to one cell. The PDU-scheduler is operated on the active flows

of the cell, i.e. with a non-empty PDU-flow-queue. As mentioned above, since systems of serially uncoupled schedulers can show undesirable behavior, both schedulers are linked together by means that the MAC scheduling is driven by the PDU scheduler's state. This is shown as clouds in figure 5 for each MAC scheduler.

In case of a handover between two schedulers not only the status information has to be transferred from the old scheduling system towards the target scheduler as described with regard to Fig. 1. The layer-3 PDUs may be rerouted to the new PDU scheduler, which happens for inter-RNC handover.

In the case of intra-RNC handover, where the PDU queue is accessible for all MAC schedulers running in this network element, the separate scheduling system may still access the already existing PDU queue without need to move the LLC data, i.e. logical-link-control data. This case is depicted in Fig. 5.

In contrast to the scheduling method having a single PDU scheduler as described with regard to Fig. 3 the scheduling method of separate scheduler per cell allows an easier synchronization of the data flows without rearranging a large PDU list. For the internal scheduling computation each scheduler treats a handover flow similar to a newly established flow in its cell. When using the invention by means of a separate PDU scheduler per cell the priority of the traffic to be compensated and hence the MAC scheduling order in the target cell is implicitly given by the MAC scheduler policy and the assignment of the transport format set as described with regard to Fig. 2.

According to the above description of preferred embodiments, one main application of the invention is the MAC level scheduling system preferably using a combination of coupled Layer 3 and MAC layer schedulers. However, the invention also is covering embodiments adapted to be used in general by scheduling systems, where the task of efficient mobility handling on MAC level has to be solved.

Moreover, as it is obvious for a person skilled in the art, the proposed scheduling approach is especially suited for scheduling data flows of a CDMA-based system or TDMA-based system or for scheduling systems in packet switched cellular mobile communication systems like UMTS or GPRS but not limited to these standards.

APPENDIX A

Method and System for UMTS Packet Transmission Scheduling on shared Downlink Channels

An improved method for packet transmission scheduling is provided, especially on downlink shared channels and an improved packet transmission scheduling system, both the improved method and the system especially adapted to be used for UMTS systems.

Accordingly, the method uses a QoS-scheduling for handling multiple data flows in a Code Division Multiple Access (CDMA) system by dynamically scheduling protocol data units (PDU) in dependence of allocated radio resource constraints, especially ensuring the required data rates due to a rate conserving scheduling by simultaneously performing an optimization of the usage of radio resources.

The QoS-scheduling is apt to handle the data flows on downlink shared channels but can also be applied to scheduling of multiple data flows for different users on a dedicated channel in the downlink direction and for a single user in the uplink direction.

The method relies on two schedulers, which are linked together in a novel manner, whereby the first scheduler provides a certain degree of predictable behavior and the second scheduler provides Medium Access Control (MAC access) and also allows for bandwidth conserving segmentation and allocation strategies.

These two schedulers are named PDU scheduler and MAC scheduler. For the basics of this scheduling method reference is made to the co-pending application filed

APPENDIX A

with the same Patent Office, Stefan Gruhl: "Method of linking two schedulers of a multi-layer network and a network comprising a transceiver having linking functionality for two schedulers", the contents of which is incorporated by reference hereby. In the present application proposal the scheduling method of the above-cited co-pending application has been adapted to the UMTS mobile communication system. It is especially shown how to link the two schedulers and parameterize the local algorithms for UMTS, especially for the medium access control (MAC) scheduler. The method is described in view of preferred embodiments in more detail below and reference is made to the accompanying drawings.

Brief Description of the Drawings

Figure 1: shows a message flow for adding a radio bearer or user equipment (UE) to the scheduler;
Figure 2: shows the allocation of channelisation codes using the code branch allocation (CBA) method in view of a preferred embodiment;
Figure 3: shows principles of the quality of service (QoS) scheduling method;
Figure 4: shows an improved medium access control (MAC) scheduling mechanism;
Figure 5: shows how to process power limits within an improved medium access (MAC) scheduler based on a preferred embodiment.

For a better understanding of the method and especially to ensure the improved performance of the method and devices, certain requirements should be met and a number of assumptions are made in advance.

APPENDIX A

Assumptions and Requirements

A certain amount $P_{PS} = \alpha_{PS} \cdot P_{max}$ of the overall maximum transmission power P_{max} is allocated by the radio resource management unit (RRM) to the packet switched radio bearer. The scheduler is apt to use P_{PS} essentially autonomously without invocation of the radio resource management unit (RRM).

If automatic repeat request (ARQ) is applied, the number of retransmissions is assumed to be significantly smaller than the regular traffic.

All transmissions having certain assigned quality requirements are embedded into a data flow. Accordingly, a data flow is defined as a sequence of data packets from the same source to the same destination in the network, for which the user has certain QoS requirements.

Each radio bearer is related to a single data flow. Because multiple radio bearers might be established for a single user, multiple data flows could exist simultaneously that are related to a single user, too. In the following description all data flows are handled separately.

Throughout this description the elements of a data flow are defined as Protocol Data Units (PDU or PDUs).

These PDU are typically layer 3 elements from an UMTS point of view, but the method is not necessarily limited to this.

Protocol data units (PDUs) are segmented into transport

APPENDIX A

blocks (TB or TBs), which receive their own headers, as specified in the UMTS 3GPP standards. This operation is associated with layer-2. Typically but not necessarily the transport blocks have a fixed size. An arbitrary number of transport blocks can be put together to one Transport Block Set (TBS). Typically but not necessarily only transport blocks of one protocol data unit (PDU) are put together.

One TBS is scheduled by the medium access control layer within the scheduling interval (typically 10ms) to the physical layer (PHY-layer) per flow.

For the downlink shared channel there is no soft handover (HO) assumed. Therefore, the scheduler handles the data flows addressed to UEs of the own cell, only.

Any mobility related procedures, e.g. hard handover are handled by the radio resource management system (RRM) independently.

The Bit Error Rate (BER) of a data flow is a static QoS requirement of the associated radio bearer. Depending on the delay constraints, there is a trade-off between Forward Error Correction (FEC), i.e. the received signal energy to noise ratio E_b/N_0 vs. automatic repeat request (ARQ) methods, i.e. the allowed number of retransmissions.

It is assumed that the required bit error rate always can be received from the core network or radio access network.

In order to optimize the bandwidth consumption of a data

APPENDIX A

flow padding is minimized as a trade-off for delay where possible. This is indicated by the flow's QoS constraints and the recent flow status.

In order to optimize the delay, the whole protocol data unit (PDU) is taken by the PDU scheduler preferably at once.

The downlink shared channel preferably is time synchronized, i.e. every data flow starts its transmission at the same point of time. Discontinuous transmission (DTX) on the downlink shared channel could lead to large fluctuations in interference to users on other dedicated channels (DCH) of the same cell or to all users of the adjacent cells. Hence, a discontinuous transmission (DTX) should not be used in downlink shared channels.

According to recent 3GPP standards there is no physical multiplexing (or PHY MuX) for different data flows in the downlink shared channel. As a consequence thereof, the transport format combination sets (TFCS) on the downlink shared channel consists of a transport format set (TFS) for one data flow, only. The transport format set is associated to the data rates R_B of the respective data flow. The transport format sets are directly related to the spreading factor SF of the Code Division Multiple Access (CDMA) transmission system which is used to support that data rate.

The transport block size within one scheduling interval remains constant for each protocol data unit (PDU). Consequently, only the number of transport blocks needs to be counted for medium access control (MAC) scheduling.

APPENDIX A

Radio Resource Allocation (RRA)

Basics on Radio Resource Allocation for a downlink shared channel

Due to its nature and in view of using a Code Division Multiple Access (CDMA) method, the main resource in the UMTS mobile communication system is the transmission power, which has to be spent for a certain user. The transmission power P_{tri} of data flow #i is expressed as

$$P_{tri} \approx \left(\frac{E_B}{N_0}\right)_i \cdot \frac{R_{Bi}}{W} \cdot \frac{I_{0i}}{h_i} = R_{Bi} \cdot C_i \qquad \textbf{equation 1}$$

where

$(E_B/N_0)_i$ denotes the to be received signal energy to noise ratio for data flow # i,

R_{Bi} denotes the current data rate used by flow #i,

W is the chip rate which chip rate at the moment is defined for UMTS as W = 3.84 MChip/s,

I_{0i} denotes the interference at the user equipment (UE) where the downlink data flow is addressed to,

h_i is the path-loss between the user equipment (UE) and Node B.

Resources in CDMA systems depend not only on the data rate as handled by conventional scheduling methods or prior art schedulers but also on a factor C which depends on several other parameters, like path-losses and interference. From equation 1 C_i is given as

APPENDIX A

$$C_i = \left(\frac{E_B}{N_0}\right)_i \cdot \frac{1}{W} \cdot \frac{I_{Oi}}{h_i} \qquad \text{equation} \qquad 2$$

The C_i-value is an essential part of the scheduling algorithm. Depending on the time when it is used, this value preferably is calculated in two ways:

At or during the establishment of the data flow equation 2 may be used directly, where $(E_B/N_0)_i$ is initially determined from the bit error (BER) requirement of the respective radio bearer and (I_{oi}/h_i) is estimated from a measurement that is signaled from the user equipment (UE) to the network.

After a communication channel is established with the radio bearer, there are no regular measurements of (I_{oi}/h_i) available in the network. Additionally, $(E_B/N_0)_i$ may differ from the initial value, e.g. due to a varying environment. Then, equation 2 should not used then, i.e. after establishing of the communication channel. At this moment equation 1 is used with

$$C_i = P_{tri}^{prev} / R_{Bi}^{prev}, \qquad \text{equation 3}$$

where P_{tri}^{prev} and R_{Bi}^{prev} are the previous transmission power and previous data rate of data flow #i, respectively.

The overall transmission power $\sum_{i \in active} P_{tri}$ of all active data flows on the downlink shared channel is limited by the allocated transmission power P_{PS} for the packet switched users. Therefore, the overall transmission limit for all active data flows is

APPENDIX A

$$\sum_{i \in active} P_{tri} = \sum_{i \in active} R_{Bi} \cdot C_i \le P_{PS} .$$ **equation 4**

Due to limited capabilities of e.g. a single channel transmission unit there is a restriction of the transmission power of a single data flow $P_{\max}^{single}$, too. Consequently, in addition to equation 4 the following limit is given for all active data flows

$$P_{tri} = R_{Bi} \cdot C_i \le P_{\max}^{single}, \quad \forall i \in active .$$ **equation 5**

Tasks and Functions of the improved Radio Resource Allocation (RRA)

Figure 1 shows a message flow between a Core Network (CN), a Radio Network Controller (RNC) and User Equipment (UE) when a new radio bearer is added to the scheduling function.

Although it is an essential part of a UMTS mobile communication system, the base transceiver station (BTS) is not separately shown as these stations are well known to persons skilled in the art. The following tasks and functions should be performed by the radio resource allocation units (RRA) before the scheduling function or operation starts:

1. Radio Bearer Establishment Request

During this phase a new radio bearer establishment is requested from the Core Network. This request must contain or specify the quality of service (QoS)

requirements of the associated data flow, i.e. the requested bit error rates (BER), data rates to be transmitted and delay requirements. If there is no radio resource control (RRC) connection established, a radio resource control connection establishment procedure between the radio network controller (RNC) and the user equipment (UE) has to be performed which is shown as step 1a in Figure 1.

2. Admission Control (AC)

The purpose of admission control is to decide whether this new request can be admitted or not. For admission control several parameters like the requested quality (QoS) of the service and the current network load are used. Other reasons for not admitting the request might be that no radio resources are available which is checked by the next step. If the request is denied there might be a negotiation procedure with a lower quality of service QoS.

3. Dynamic Channel Allocation

The Dynamic Channel Allocation (DCA) procedure allocates the following transmission parameters to the data flow (non exclusive): Transport format set (TFS), radio link control (RLC) Info, new Channelisation Code, Initial transmission power etc. For the allocation method for transport format set and channelisation code see also paragraph „Allocation of Data Rates" of this description. A new amount of transmission power P_{packet} for the scheduler can also be allocated by DCA.

4. Radio Bearer Setup

This function performs a setup of the Radio Bearer and

synchronization between RNC and UE. Additionally, the BTS will be initialized with the parameters that are allocated by DCA which is not shown in Figure 1 but known to a person skilled in the art.

5. Start Dynamic Scheduling

After a successful establishment and initialization the new data flow is added to the scheduling function. The scheduling function will be performed now also for this flow. Reference is made to Figure 1 showing the message flow for adding a radio bearer to the scheduler.

Allocation of Data Rates, TFS and Channelisation Codes

Allocation of Data Rates

The allocation of the data rates for each data flow has a strong impact on the system efficiency that the scheduler can achieve. The data rates relate to TFS and the channelisation codes. According to the recent 3GPP standards there is no physical multiplexing (or PHY MuX) for different data flows in the downlink shared channel. As a consequence thereof, the transport format combination sets (TFCS) on the downlink shared channel consists of a transport format set (TFS) for one data flow, only. The transport format set is associated to the data rates R_B of the respective data flow. The transport format sets are directly related to the spreading factor SF of the Code Division Multiple Access (CDMA) transmission system which is used to support that data rate.

For a rough allocation or estimation of the limit data rates the following rules are applied:

For the maximum data rate R_{Bmax} transport format sets (TFS)

APPENDIX A

should be allocated to allow data rates up to two to four times of the maximum data rate, i.e. (2 to 4)R_{Bmax}. There are two reasons for this requirement. The first is that these maximum transport format sets are required from the MAC-scheduler to serve a flow with a temporarily higher data rate than requested to let a flow profit from other flows being idle. This applies when there is remaining capacity on the air-link and if this flow has already sent packets to the PDU list ahead of its specified rate.

The second reason is to allow for a time divisional multiplex style multiplexing on the transport block level.

The MAC-scheduler's algorithm can be developed towards bandwidth efficiency. Therefore it also may be desirable to use arbitrary sizes of transport blocks to minimise padding. This means the available transport format should be able to temporarily exceed specified rates.

Fairness, bandwidth and quality of service (BW-QoS) guarantees among flows are maintained by another scheduler, the PDU scheduler.

For the minimum data rates R_{Bmin} transport format sets should be allocated to allow data rates below R_{Bmin}. The availability of smaller transport format sets allows the MAC-scheduler to minimise padding. As this introduces additional delay and lowers the average transmission rate, it is only applicable to certain QoS flows. The optimization of such allocation and TFC usage is addressed separately.

APPENDIX A

In order to comply with the assumption that the downlink shared channel preferably is time synchronized, i.e. every data flow starts its transmission at the same point of time, and discontinuous transmission (DTX) on the downlink shared channel could lead to large fluctuations in interference to users on other dedicated channels (DCH) of the same cell or to all users of the adjacent cells, only those transport format sets are allowed which would fill up the whole data frame with data. Because the spreading factors of the channelisation codes are in the order of $SF = 2^k$, $k = 2,3,\ldots$, according to the aforementioned recent 3GPP standards, this leads to data rates of $R_B = R'_B \cdot 2^n$, $n = 0,1,\ldots$, where R'_B represents a reference data rate for a certain given spreading factor and may become R_{Bmin}.

<u>Transport Format Set (TFS)</u>

The transport format set is defined as the set of transport formats that are associated to one data flow. The semi-static part (coding, transmission interval, rate matching) essentially determines the bit error rate. It is defined by the radio resource management. In the following discussion, the focus is only on the dynamic part which consists of the transport block size and the transport block set size. The dynamic part of transport format sets can be used for optimization of the segmentation in the RLC. For the choice of this dynamic part there is a trade-off between the granularity of the data rates and the limited size of transport format sets. On the one hand, each data flow intends to have a high granularity in data rates avoiding extensive padding. This would lead to a large transport format set size. On the other hand, a transport format set is used to enable efficient physical or PHY signaling for changing data

APPENDIX A

rates. Due to limitations of that PHY signaling (e.g. TFCI (Transport Format Combination Indicator) encoding) the maximum transport format set size is quite limited. Therefore the following transport format set allocation rules regarding the characteristic of the data flow are proposed and used according to the method:

1. Real Time (RT) Services: This service type needs immediately serving of the offered data. Hence, a high granularity towards higher data rates is desired. Therefore, for real time services a larger transport format set should be allocated.

2. Non Real Time (NRT) Delay Sensitive Services: Here, a limited automatic repeat request (ARQ) can be used for protection of the data flow. The granularity is not as high as for the pure RT service because some data can be queued for a limited time. Therefore a limited transport format set can be allocated for such services. For efficient use of automatic repeat request mechanisms the transport block size should be small.

3. NRT Unconstrained Delay Services: This type of service is the best candidate for bandwidth optimization. In principle unlimited queuing is possible. Hence, no much granularity is necessary. Therefore a quite limited transport format set can be allocated for this service type. Granularity is used to avoid padding, only.

Beyond the delay constraint it is sensible to take further QoS requirements and flow specifications into account. The system could potentially adopt to certain preferred PDU sizes, like that of a transport control protocol acknowledgement (TCP-ACK). Bulk data transfer

APPENDIX A

could guard the selection towards max. PDU size, etc.

DL Channelisation Codes

For allocation of downlink (DL) channelisation codes the method of Code Branch Allocation (CBA) as described in Qiang Cao, Seau Lim, Jens Mueckenheim: „Code branch allocation for CDMA systems," Patent Appl. EP 99 301 810.0, filed 10.03.99 is used. The CBA method is an attempt to solve the code space shortage problem, which is especially relevant in the downlink. CBA defines a path in the code tree, the code branch, which consists of spreading codes for each SF that can be used for the transmission. The code branch might be transmitted to the UE. There can be intersections of code branches where only exclusive use of one code branch simultaneously is allowed. Because there is a fixed relation between data rate and spreading factor (SF), see also the above paragraph assumptions and requirements, an allocation rule of the path in the code tree is used in view of recent 3GPP standards. According to recent 3GPP standards there is no physical multiplexing (or PHY MuX) for different data flows in the downlink shared channel. As a consequence thereof, the transport format combination sets (TFCS) on the downlink shared channel consists of a transport format set (TFS) for one data flow, only. The transport format set is associated to the data rates R_B of the respective data flow. The transport format sets are directly related to the spreading factor SF of the Code Division Multiple Access (CDMA) transmission system which is used to support that data rate. The following allocation rule is used according to the method:

1. For data rates below the maximum requested one the nodes of the code branch should be allocated always

APPENDIX A

below intersection points. This allows guaranteed data rates R_{Bmax} for all data flows, if no padding-reduction strategy is used in the MAC-scheduler. For flows that have stringent BW-QoS requirements and where still padding-reduction is desirable, CBA should only be made with unconstrained flows.

2. For TFC for higher data rates of 2 to 4 times$\cdot$ R_{Bmax} that can be used for the proactive scheduling (cf. data rate allocation rule 23) nodes in the code tree can be used that are above intersection points. Because only a quite limited number of data flows is allowed to use this larger data rates no conflicts in the channelisation code usage are expected.

In the following usage of the CBA method for allocation of channelisation codes for the scheduling on downlink shared channel is explained by means of an example. Figure 2 illustrates a code tree, which is a representation of the orthogonal variable spreading factor codes that are used on the UMTS downlink shared channel. Every node characterizes a code sequence with a spreading factor given by the first number. All codes within the code tree cannot be used simultaneously. A node can be used for a physical channel if and only if no other node on the path from the specific node to the root of the tree (i.e. with lower SF) or in the sub tree below the specific node (i.e. with higher SF) is used by another physical channel. In the example it is assumed that the sub tree below node 4,1 is reserved for downlink shared channel usage. The assignment of the nodes for the code branches to two users can be as follows:

APPENDIX A

Because the nodes below node 8,1 and 8,2 can be used simultaneously, these sub trees can be assigned for the data rates below R_{Bmax}. For example, user #1 gets the nodes 8,1 and 16,1. User two will be assigned to nodes 8,2 and 16,4.

For the proactive scheduling purpose node 4,1 is allocated to both, user #1 and user #2. They cannot be used simultaneously. Hence, the scheduler must ensure that when user #1 takes node 4,1 user #2 should not transmit on any node of its code branch and vice versa.

Hence, the code branch for user #1 is: (4,1); (8,1); (16,1) and for user #2: (4,1); (8,2); (16,4), see also Figure 2.

The Scheduling Method

The method contemplates using of two schedulers, which are linked together in a novel manner to achieve a certain degree of predictable behavior, while also allowing for bandwidth conserving segmentation and scheduling, see also Stefan Gruhl in "Method of linking two scheduler's of a multi-layer network and a network comprising a transceiver having linking functionality for two schedulers". These two schedulers are named PDU scheduler and MAC-scheduler.

The first one operates on the input data from Layer 3, the Protocol Data Units (PDU). It receives the QoS requirements of each flows and determines the order in which PDUs should receive service. This service is delivered by lower layer protocol functionality and incorporates mainly two steps. The first is Layer-2 protocol related and includes mainly segmentation of PDU

APPENDIX A

to Transport Blocks (TB) and Automatic Repeat Request (ARQ).

The MAC-scheduler serves the PDU's from this list and tries to reflect the order in the list, while also taking CBA-, timing and power constraints into account.

In Figure 3 the principle architecture of the two serial schedulers is shown. The MAC-scheduler is active at every frame, e.g. on a 10 ms base. The PDU scheduler is operated on all active flows, i.e. with a non-empty PDU flow-queue.
Systems of serially uncoupled schedulers can show undesirable behavior. Therefore, both schedulers are linked together.

To allow for the linking the intermediate protocol functionality (depicted as cloud in Figure 3) has to allow for a certain implementation summarized below. It is assumed that layer-2 protocol functionality is apt to be processed as a stateless operation with the character of a function with negligible processing time. Two main tasks in layer-2 are identified which are for the sender, the segmentation process to transport blocks TBs and an ARQ stage for transport blocks TBs. The segmentation process is intuitively a stateless function as it turns one input element in an output vector of new elements (typically with new headers for sequence numbers, etc...). Automatic repeat request ARQ is processed as a function with the input of a TB and the output of the same TB, with ARQ parameters set. Besides this output there is potentially other asynchronous output of the ARQ stage:

a) If the ARQ window size is reached, no output is

generated until former transmissions were successfully acknowledged by the receiver. This case can be neglected if always a test on the available window size is done before performing the ARQ operation.

b) When the automatic repeat request (ARQ) stage receives the request for a retransmission, it generates an output without input.

The number of retransmissions is assumed to be significantly smaller than the regular traffic and therefore treat this relatively rare case as a separate ARQ-process which generates its own traffic regardless of the traffic which is mainly running through the ARQ stage in the describe function manner.

c) If the maximum number of retransmissions is exceeded and the ARQ considers a PDU transmission unsuccessful, there is another asynchronous signal that has to be signaled to upper layers.

The central idea to allow for such linking of the two schedulers is that the MAC-scheduler operates on PDU's in the Protocol Data Unit (PDU) list, while this list is dynamically changed by the Protocol Data Unit (PDU) scheduler. As a consequence, the whole PDU access has to take place via reference to allow for protocol operation on demand techniques. This has to be accompanied by locking of the shared element, the PDU list.

The exact linking method and further details are subject of the above-cited application from Stefan Gruhl: "Method of linking two schedulers of a multi-layer network and a network comprising a transceiver having linking

functionality for two schedulers".

PDU Scheduler

The PDU scheduler takes the PDU from the incoming data flows. Each data flow is queued in its own FIFO-queue, denoted as PDU flow-queue. They are scheduled regarding their QoS requirements into one common list for the MAC-scheduler. This list is denoted as PDU list. This list is not termed a queue, because due to MAC-constraints it cannot be assured to serve this queue in a FIFO fashion. The PDU scheduler must be able to serve the data with the required data rates. For this purpose any rate conserving scheduling policy can be applied, see Hui Zhang, "Service Disciplines for guaranteed Performance Service in Packet-Switching Networks," Proceedings of the IEEE, Vol. 83, No. 10. October 1995, e.g. Weighted Fair Queuing (WF^2Q) or Virtual Clock Queuing (VCQ).

The scheduling elements for this scheduler are taken depending on the following rule: Regularly the PDU's are large enough to be scheduled as one unit. In this case one scheduling element equals one PDU.

If it is possible to serve several PDU's simultaneously on the MAC-layer it may become desirable to have several PDU's available for MAC-scheduling from one flow. This would mainly be the case where PDU's become too small to be served with the required minimum data rate R_{Bmin} within the given MAC-scheduling interval $T_{schedule}$ (typically $T_{schedule}$ = 10ms), i.e. when

$$PDU_{len}/R_{B\min} \le T_{schedule} \qquad \textbf{equation 6}$$

APPENDIX A

The problem can be overcome by having several PDU's from the flow grouped together into one container, which then becomes the scheduling element.

Hence, the scheduling element can be defined as one container that may consist of one (normally) or several PDU's. Throughout this document one scheduling element is defined as PDU and the term PDU scheduler is used for the sake of simplicity.

As shown in J. Cobb, M. Gouda and A-El-Nahas, "Flow timestamps," Annual Joint Conference of Information Sciences, 1995 (Appendix C), it is also reasonable to work with flow time-stamps instead of PDU time-stamps. When doing so, the inventive PDU scheduler becomes active when a PDU from a flow is fully served and therefore removed from the PDU list or when a formerly inactive flow gets reactivated by a PDU arrival into its empty PDU flow-queue. This is beneficial as it limits the number of elements in our PDU list to the number of active flows.

MAC-Scheduler

Main Function of the MAC Scheduler

The MAC-scheduler serves the PDU's from the PDU scheduler. The order in the PDU scheduler's list signals the priority in which the PDU scheduler wants the PDU's to be served. The MAC scheduler tries to achieve this while obeying three restrictions:

Bandwidth constraint due to TFC allocation of the flow and availability of a node in the code tree (refers to CBA)

Delay constraints:

Drives the decision how many subsequent TBS transmissions spread over several timing intervals are tolerable to

obey timing requirements of the served PDU

ARQ-constraints: Transmission of TB's that receive ARQ service is only possible until the ARQ window size is reached. Further transmissions are possible only after the ARQ stage receives the acknowledgement from the receiver.

Power constraints:

Both the power for the transmission to one individual mobile and the overall power in the cell is limited. To avoid RRM regulation on these issues, the scheduler should itself take this into account.

This present proposal mainly consists of a framework that allows for several MAC-scheduling algorithms to obey these constraints without having to explicitly worry for the flow's QoS requirements anymore, as this has been applied by the PDU scheduler already.

In the following an algorithm is used complying with these constraints in a straightforward manner. Later some improvements are shown.

The basic mechanism for MAC scheduling is depicted in Figure 4.

It essentially is the following:

1. Set the queuing pointer in front of the PDU list, i.e. set pointer = 0. Reset the total consumed power $P_{current}=0$.

2. Take the next PDU from the PDU list and consider as much of it in means of TB for scheduling as it is constrained by:

   PDU_size / segment size → result max #1 of TB's

APPENDIX A

ARQ-constraint → result max #2 of TB's

TFC-constrain → result max #3 of TB's

Maximum single transmission power P_{max}(single):

R_{Bi}(max) =

P_{max}(single) / C_i (cf. equation 5), where C_i is currently

given by equation 2 or equation 3

→ result max #4 of TB's

Overall power limit P_{limit}: Calculate the virtual available data rate R_{Bi}(available) = (P_{limit} − $P_{current}$) / C_i (cf. equation 4) → result max #5 of TB's

Formal: Result $\#TB_{max}$ = min (max #1 of TB's … max #5 of TB's);

3. Perform capacity optimizing decisions on TBS creation. It can be advisable to schedule less TB's, with a smaller TBS than possible from the constraints in step 2. If no optimization is desired, chose max #of TB's from step 2.
   → new variable $\#Tb_{schedule}$

4. Create $\#TB_{schedule}$ TB's. Therefore the segmentation and ARQ on demand is "executed" on them. The such created TB's are assembled to a TBS. Set R_{Bi}(used) regarding the created $\#TB_{schedule}$.

5. Take the TBS and store them together with the associated TF for delivering to PHY-layer in step 8.

6. Compute the total cell power by $P_{new} = P_{current} + C_i \cdot R_{Bi}$(used). Compare this value against a power limit P_{limit}.

7. If total Power check is ok, i.e. if $P_{limit} - P_{new} \geq P_{min}$

APPENDIX A

(P_{min}: minimum transmission power for a certain # of TB's) and there are more PDU's in the PDU list, increase P_{cell} by one to the next PDU in the PDU list, set $P_{current} = P_{new}$ and go to step 2.

8. Deliver all stored TBS together with their associated TF to PHY layer.

Handling of the power limit P_{limit}

This section describes how the power limit P_{limit} for the cell is allocated for MAC-scheduling. The power limit P_{limit} for the scheduler should be chosen according the following rule:

$$P_{limit} = \min\{P_{PS}, P_{current}(t-1) + \Delta P_{inc}\} \qquad \textbf{equation 7}$$

The first term in equation 7 prevents the scheduler to use resources larger than the by RRM assigned P_{PS}. The second term guarantees that the increase in the current transmission power $P_{current}$ is below a given limit ΔP_{inc}. This limitation is useful in order that the DL power control for all users on other TrCH which are not handled by the scheduler (e.g. users on DCH and of the adjacent cell) can follow the increase of transmission power on downlink shared channel. The overall downlink shared channel transmission power from the previous time $P_{current}(t-1)$, which is estimated by the sum of the transmission powers P_{tri}^{prev} of all previous active data flows #i

$$P_{current}(t-1) = \sum_{i \in prev\ active} P_{tri}^{prev} \qquad \textbf{equation 8}$$

APPENDIX A

P_{tri}^{prev} (also used in equation 3) is simply the transmission power of the code sequence that was associated to data flow #i (Code Tx power). It can be measured in time periods of some scheduling intervals (up to 100 msec) which is much faster than the overall transmission power measurement (up to 1sec) that is used by RRM.

The basic limit definition in equation 7 can be enhanced as described in the following. For the purpose of efficient handling of the allocated radio resources the MAC Scheduler should monitor the goodput, i.e. the throughput R_{actual} of the scheduler without retransmissions, which simply is defined by

$$R_{actual} = \sum_{i \in active} R_{Bi} \qquad \textbf{equation 9}$$

The virtual bandwidth is defined by the overall available data rate $R_{overall}$, which can be allocated by the MAC Scheduler. This virtual bandwidth depends on the allocated transmission power P_{PS} for the scheduler:

$$R_{overall} = function(P_{PS}) \approx P_{PS} / C' \qquad \textbf{equation 10}$$

The value of C' represents a kind of estimate from the constants C_i from all data flows. The goodput R_{actual} is now compared with the virtual bandwidth $R_{overall}$. Depending on the comparison result the following actions can be taken:

If $R_{actual} < R_{overall}$, then there is a scheduling problem. The scheduler can process less data than required. In this

APPENDIX A

case RRM has to be informed to take actions. This could involve to allocate larger resource P_{PS} to the scheduler if available. If not, a dynamical resource reallocation to flows has to be performed. This could for example mean to drop or stop certain flows, that formerly were served with QoS BW guarantee. Finally this feedback can be used to alter capacity estimates for future Admission Control decisions.

If $R_{actual} \approx R_{overall}$, then the scheduler works efficient and within the limits. In this case equation 7 will be used as scheduling policy.

If $R_{actual} >> R_{overall}$, then the scheduler works in a relaxed manner. That means it is able to schedule much more data than actual required. In this case the scheduler can have a self limiting behaviour depending on the history of the goodput R_{actual} in the following way:

if $R_{actual}(t) \leq R_{actual}(t-1)$, then use the following modification of equation 7:

$$P_{limt} = P_{current}(t-1) - \Delta P_{dec} \qquad \textbf{equation 11}$$

where ΔP_{dec} is a certain decrease of transmission power.

if $R_{actual}(t) > R_{actual}(t-1)$, then use equation 7 as it is. This allows equalisation of the total traffic in terms of traffic shaping. In order to keep the allocated resources available for the scheduling process RRM will not be informed about this self limiting. Nonetheless there will be a noticeable lower variation of power consumption on

APPENDIX A

downlink shared channel neighbouring cells and DCH	which is beneficial for power control fluctuation in this cell.

Figure 5 illustrates an example of handling P_{limit}. Both, the usage of the increasing limit ΔP_{inc}, when more power is required, and the usage of the decreasing limit ΔP_{dec} for less required Tx power are shown.

Improvement of MAC-scheduling decision

Especially for NRT services it is not necessary to always attempt to pack the whole PDU into one TBS to be scheduled in one MAC-scheduling interval. It might be desirable to spread the transmission in time over several scheduling intervals. Hence, the method proposes and uses the following enhancement to section main function:

0:

For NRT-services for each PDU the maximum number $N_{schedule}$ of MAC-scheduling intervals $T_{schedule}$ that are allowed for an initial PDU transmission is determined. "Initial" means that this value does not include potential retransmissions. The value $N_{schedule}$ is determined by:

$$N_{schedule} = PDU_{len} / \left(R_{B\min} \cdot T_{schedule} \right) \quad \textbf{equation 12}$$

Here, a system is assumed that obeys no other restrictions (e.g. ARQ..., see assumption mentioned above that if automatic repeat request (ARQ) is applied, the number of retransmissions is assumed to be significantly smaller than the regular traffic). Given this value for each PDU, the MAC scheduler can schedule less TB at a time to reduce the padding. This is possible because the remaining data are apt to fit into a smaller TBS in the

next interval.

Conclusion

The method is not restricted to the specific embodiments described above. A person skilled in the art will recognize that based on a rate conserving strategy the scheduler guarantees the required data rates.

Even if a delay is not explicitly addressed by the scheduling principle, if the respective data flow is in compliance with its required quality of service the scheduler guarantees that there will be no additional delay due to congestion within the scheduling system.

Preferably, bit error rate (BER) requirements are guaranteed in addition via properly forward error correction (FEC) and automatic repeat request (ARQ) functions.

An application of the improved scheduler is a handling of data flows in the downlink shared channel and the downlink shared channel scheduling was described in detail in the above description. However, the inventive method of QoS scheduling is not limited to downlink shared channels. It also may be applied to scheduling of multiple data flows for different users on dedicated channels (DCHs) in the downlink (DL) and for a single user in the uplink (UL).

Bernhard 3-8-14

APPENDIX A

Fig. 1

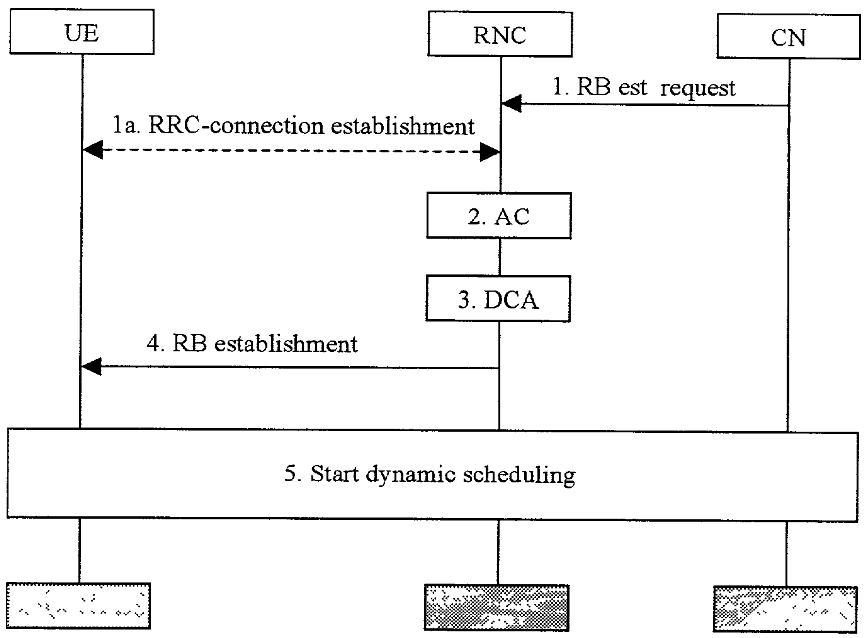
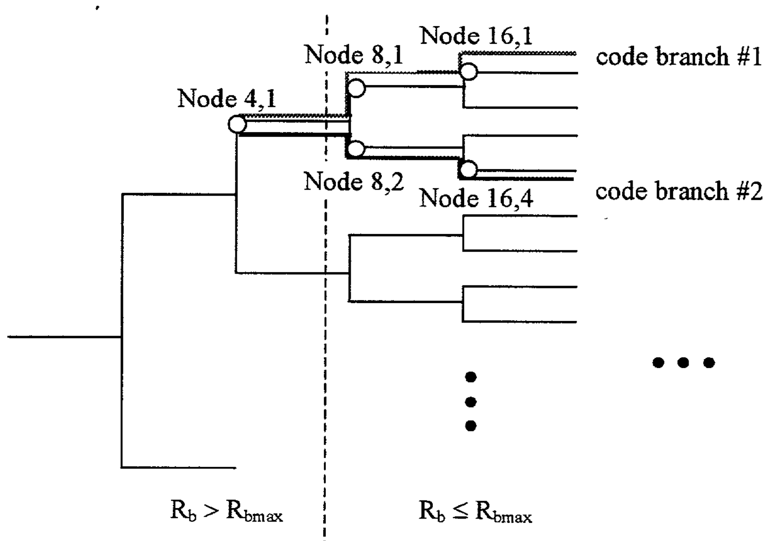


Fig. 2



APPENDIX A

Fig. 3

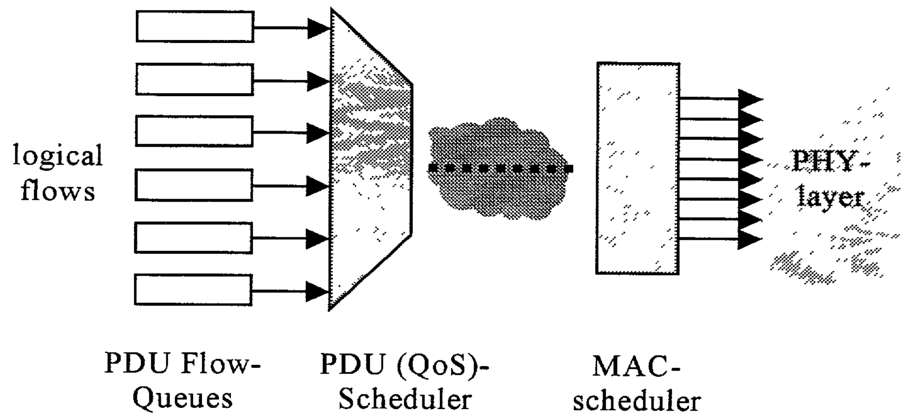
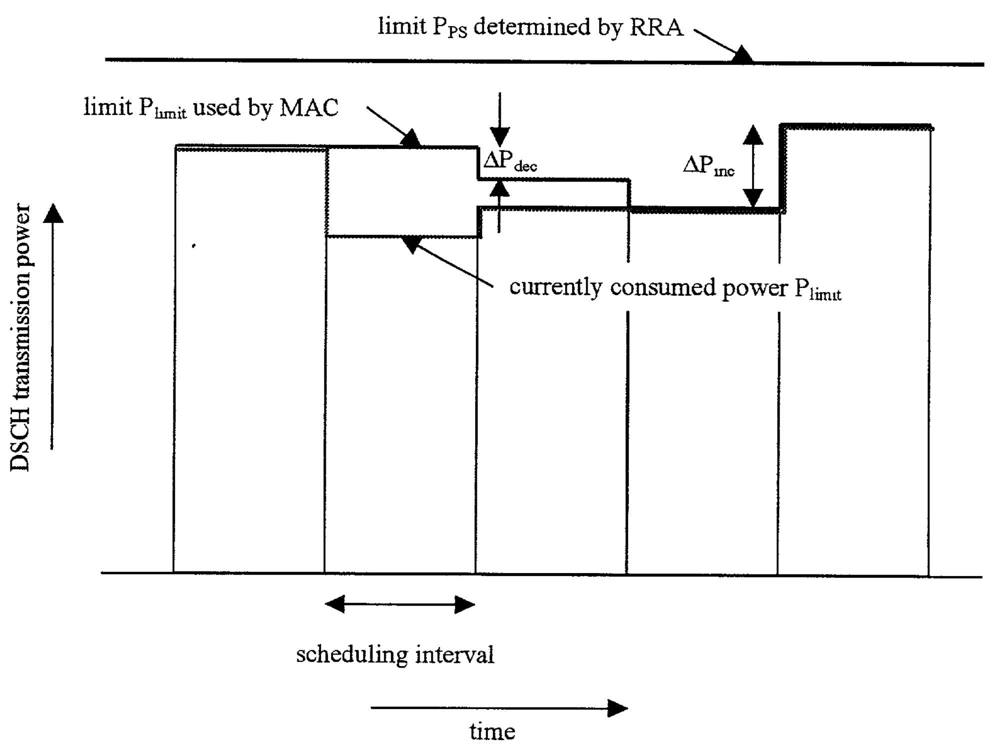
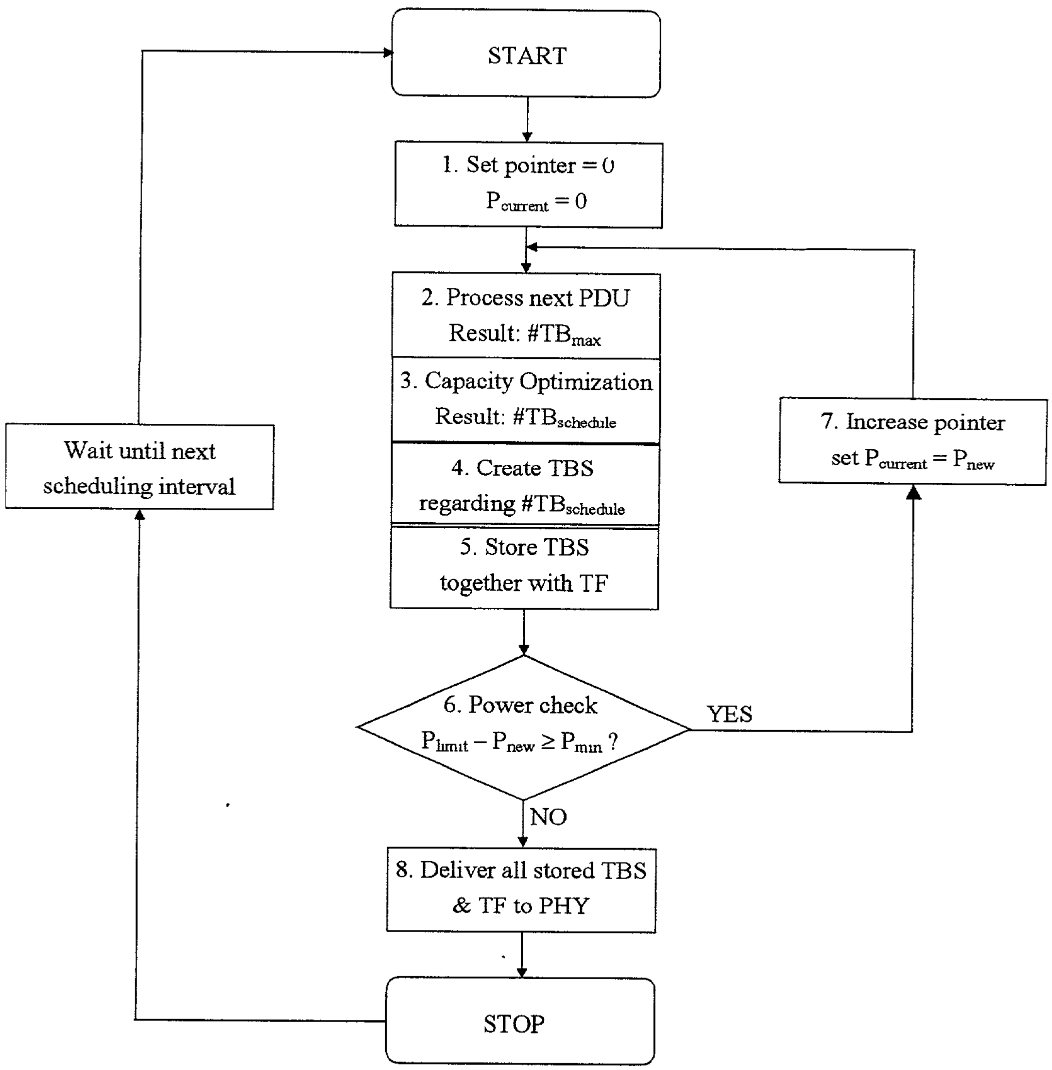


Fig. 5



APPENDIX A

Fig. 4



APPENDIX B

Method of linking two schedulers of a multi-layer network and a network comprising a transceiver having linking functionality for two schedulers.

The method mainly addresses the problem of multi-layer scheduling in a packet switched network or system, for example in a mobile telecommunication system. Considering packet-switching networks with the standard ISO/OSI architecture the task of multiplexing essentially reduces the task of ordering packets and then sending them serially over a shared link. This process of serialisation is subsequently referred to as scheduling.

For certain network links, especially for wireless links an certain amount of pre-given link characteristics have to be applied. Such a pre-given link characteristic may be for example a very specific error character of the link. This is traditionally addressed by applying segmentation on the layer-3 Protocol Data Units (PDU´s). Then after segmentation the resulting smaller entities are scheduled in a link specific medium access layer. Since, however, scheduling of protocol data units have been already performed on the higher layer there are two subsequent schedulers resulting in numerous disadvantages if these two schedulers are de-coupled.

Typically there are several protocol layers on top of each other and each of those layers operate on protocol data units (PDU). Based on the data flow within the ISO/OSI standard layer model for protocols, the output of one layer usually provides the input of the next layer. This model for protocols seems to be beneficial since it allows several functions to be grouped together into dedicated modules. If, however, a protocol function contains scheduling this standard approach is less

APPENDIX B

desirable when the dynamic behaviour of subsequent functions in subsequent layers is not predictable. Moreover for persons skilled in the art it is obvious that it is often impossible or not advisable to restrict the providing of scheduling processes for the whole protocol-stack to only one scheduler.

In general an arbitrary number of individual logical data flows is given forming the bases for providing Quality of Services (QoS) by means of the individual flows, each of which having a set of QoS-attributes associated. According to the document "Service disciplines for guaranteed performance service and packet-switching networks" issued by Hui Zhang in proceedings of the IEEE, volume 83 No. 10, October 1995 scheduling algorithms are proposed to provide bandwidth conserving scheduling disciplines for the flows regardless of the packet arrival patterns of the flows. The algorithms schemes disclosed therein assume that there is exactly one shared output link with static capacity. However this view is not always applicable regarding for example wireless links with multi-slot mobile in General Packet Radio Systems (GPRS) or via a downlink shared channel in Universal Mobile Telecommunication Systems (UMTS).

Thus to address the specific character of the link, there is a lower link layer protocol layer that typically comprises segmentation to transmission blocks. The length of such blocks is not static but varies to allow the usage of different coding schemes that cause to improve the operation at a certain channel condition and an experienced Bit-Error-Rate (BER). Each block receives a header for re-assembly, a Cyclic Redundancy Check (CRC) for error dedication and possibly an Automatic Retransmission Request (ARQ) for backward error

APPENDIX B

correction. A so called MAC- scheduler (medium access control-scheduler provides these blocks for the Physical-layer (PHY-layer). In case of wireless circumstances the link is a shared medium so that the PHY-layer typically allows for several channels. However, typically there are transmission-(or radio-) blocks that can not be scheduled to any of those channels.

It is an object of the method to provide an improved possibility to overcome the above mentioned restrictions and to allow for very efficient scheduling.

The method proposes and uses a scheduling decision approach of linking two hierarchically structured schedulers by service processing on demand. By providing by means of the scheduler of an upper layer protocol data units of an incoming data flow within a priority order of a definable data-flow-queue to be served by the scheduler of a lower layer for data transmission, selecting a protocol data unit on demand of the scheduler of the lower layer in dependence of the priority order and actual network constraints, and serving the selected protocol data unit by the scheduler of the lower layer, the scheduler of the upper layer can dynamically change the elements within a definable data-flow-queue but the scheduler of the lower layer decides the further processing order with regard to provided resources that can be filled up with data units of the upper layer.

Thus, the upper scheduler advantageously is apt to read and write in a data-flow-queue and the scheduler of the lower layer is apt to read in the data-flow-queue. The upper scheduler provides a predictable behaviour in that the protocol data units queued in a respective flow have

APPENDIX B

constraints and quality-of- service-attributes associated that can be compared on demand of the lower scheduler with the actual network constraints starting with the protocol data unit having the highest priority within a respective flow-queue.

For linking the two schedulers the selected data units are delivered to a protocol functionality means for performing data unit operations necessary for data transmission.

Preferably, the method relates to two serial schedulers of a GPRS-system, i.e. the upper scheduler is operating on the Logical-Link-Control-layer and the scheduler of the lower layer is operating on the Medium-Access-Control-layer so that an implemented intermediate protocol functionality is the Radio-Link-Control-layer.

The method is described in view of preferred embodiments in more detail below and reference is made to the accompanying drawings.

In the drawing:

Fig. 1 shows the principal architecture of two serial schedulers with an intermediate protocol functionality according to the method;

Fig. 2 shows a black-box interpretations of protocol functionality;

Fig. 3 shows a referencing model of PDU-input data into an abstract protocol function;

Fig. 4 shows a general example of a two-scheduler access on a shared queue;

Fig. 5 shows a regular MAC-service to a PDU; and

Fig. 6 shows a completion of a MAC-service to a PDU according to Fig. 5.

APPENDIX B

The method contemplates using of two schedulers, which are linked together in a novel manner to achieve a certain degree of predictable behavior, while also allowing for bandwidth conserving segmentation and scheduling. In the following description, the essential working mechanism of the schedulers are described basing on a network or system having a transceiver unit operating on at least two different protocol layers, i.e. an upper layer and a lower layer each of which having a scheduler. The scheduler in the upper layer operates on any Protocol Data Units PDU's and is called PDU-scheduler. Thus the input of this scheduler are simply called PDU. The output of this upper layer's scheduler is a service list of these PDU's. According to the exemplary embodiment, the lower layer's scheduler is assumed to be the last one in the chain of schedulers. This is for the most protocol applications the MAC-layer and the lower layer's scheduler is therefore referred to as MAC-scheduler for distinguishing reasons as depicted in Fig.1.

However, for a better understanding of the subsequent description of the method and especially to ensure the improved performance of the inventive architecture, method and devices, certain requirements should be met and a number assumptions are made in advance.

Regarding two schedulers, an intermediate protocol functionality has to be implemented there between. It is assumed that the functionality allows for a view of input and output traffic plus relatively rare asynchronous output. For the following description the protocol functionality only in the direction from a higher to a lower layer is regarded as being necessary. The

APPENDIX B

interpretation of such a protocol functionality is shown in Fig. 2. As can be seen any protocol functionality is assumed to deal wit one input and a plurality of outputs. The output is assumed to have the character of one main output stream that typically is forwarded to a Service Access Point (SAP) of a lower layer. However, an additional asynchronous output may also be provided. Such an asynchronous output may for example result from an Automatic-Retransmission-Request-stage (ARQ-stage) and is sent somewhere else. Thus the asynchronous output may be sent to the former SAP in the case of a retransmission or somewhere else to a higher layer to signal an unsuccessful transmission. However, it is assumed that this asynchronous traffic should be relatively rare and that it is not required to have this output explicitly associated with the input traffic.

The following description specifically considers a system with segmentation and backward error correction functionality and the subsequently described preferred example of embodiment may be implemented in the black-box fashion according to FIG. 2.

The time of computation of such approximated functionality is small enough to allow the execution on demand. Therefore, it is necessary to be able to give an upper bound t_{max} on processing time for the maximum number of possible executions of such on-demand requests. The assumption is given in case the t_{max} is lower than one lower-schedulers scheduling cycle.

It has to be noted that the working with flow time stamps, as argued for example by J.Cobb, M.Gouda and A-EL-Nahas in "Flow timestamps", Annual Joint Conference of Information Sciences (Appendix C) and discussed in detail

APPENDIX B

below, eases the estimation of processing time since the upper limit of elements in the service-list of a PDU-scheduler can be limited to the maximum number of flows in the system.

It is assumed that all QoS traffic is embedded into flows. A flow is a sequence of packets travelling from the same source (application) to the same destination application, for example from one local Base Transceiver Station (BTS) to a Mobile Station (MS) or vice versa. For one flow there are QoS-attributes negotiated between the network and the application. A set of constraints which is also associated with the flow, might be within the network to provide such service. The number of flows in the system may be arbitrary but finite.

Throughout the subsequent description the elements of a data flow are referred to as Protocol Data Units (PDU) typically being layer-3 elements. However, since this is not necessary the protocol data units are not limited to these elements.

In the following description the segmented PDU's are called blocks. and it is always assumed that the segmentation process automatically attaches the required header and performs CRC and ARQ on it. Thus for the sake of simple description a block is considered ready for transmission once it is created.

Moreover, it is assumed that a higher layer scheduler is called PDU scheduler. This QoS-scheduling is implemented by a rate conserving serving discipline and typically is implemented by assigning service priorities to the packets. Any kind of scheduling algorithm yields an execution order for its processed packets. Thus this

APPENDIX B

order may be interpreted as an ordered list of execution as it is foreseen by the algorithm. Throughout the following description, the assumption of one ordered list is sufficient regardless of the actual serving discipline. This list is referred to as service-list.

Having regard to the principle architecture of Fig. 1, the PDU-scheduler takes the PDU from the incoming data flows. The PDU's of each data flow are queued in an own first input first output (FIFO)-queue per flow denoted as PDU-flow-queue. Each such flow has constraints and the QoS-attributes associated.

The PDU-scheduler targets certain QoS goals, as for example a certain bandwidth as it is usually targeted by rate conserving strategies. Therefore it orders the PDU's in such a way, that each flow receive service not worse than a link with the specified bandwidth. This is referred to as rate/bandwidth conserving scheduling.

A most important goal of such algorithms is to grant each flow access to the shared medium not worse than the required bandwidth and to fairly share unused bandwidth among them. This is only possible if the resource allocation did not allocate flows with more bandwidth requirements than it can be served by the shared link. However, the details on PDU-scheduling is known by persons skilled in the art and thus is not described in detail.

The PDU-scheduler experiences that some of it's scheduled PDU's are getting serviced and after a while are removed from the PDU-flow-queue and from the service-list.

However, since classic schedulers expect that always the

APPENDIX B

first element is getting full service and thus that there is a strict FIFO service manner on it's queue the PDU-scheduler has to be improved according to the method to be applicable with regard to the MAC-constraints.

With regard to a PDU-scheduler according to the method, PDU-timestamps or flow-timestamps as disclosed in the aforementioned reference "Flow timestamps", the contents of which is incorporated by reference hereby, could be used. Assigning a timestamp to every PDU by using PDU-timestamps could be performed on PDU arrival resulting in a function's timing associated with each PDU-arrival. As an alternative a timestamp could be assigned to every PDU arriving within a MAC-interval. As shown in the aforementioned and referenced citation "Flow timestamps" it is sufficient to maintain one timestamp per flow, i.e. for the first packet in the queue of each flow. As long as the QoS policy of the system should not account of the inter-arrival patterns of the PDU's the maintaining of one timestamp per flow decreases the effort significantly and has no disadvantages. However this is typically not desired as the flow should be treated with regard to it's flow specification and not it's actual observed dynamical behaviour. Also both schemes can be used within the scope of the method and for the following description it is assumed that flow-timestamps are used and consequently timing issues for flow-timestamps are designed.

As a consequence two situations where scheduling action is performed could be identified:

a) A PDU from a flow has received full service by lower layers and is thus removed from the service-list and the PDU-flow-queue. The next PDU from this PDU-flow-queue, if

available, is then assigned a timestamp and gets ordered into the service-list. Accordingly the timing of the PDU-scheduler is driven by the serving lower layers and if there is no more PDU available from the respective flow, the flow is considered to be inactive.

b) If a new PDU arrives for an empty PDU-flow-queue, this flow becomes active and needs its PDU to be scheduled, i.e. inserted into the ordered service-list. For some scheduling strategies this flow reactivation requires some flow synchronisation to be performed as known by someone skilled in the art. The timing of this event is driven on PDU arrival.

In contrast to the aforementioned PDU-scheduler a MAC-scheduler serves the PDU's from the PDU-scheduler. The order in the PDU-scheduler's queue signals the priority in which the PDU-scheduler wants the PDU's to be served. The MAC-scheduler tries to achieve this priority order while obeying restrictions. The MAC-scheduler takes the PDU's from such a queue successively, performs segmentation into transport blocks, handles the ARQ and delivers the transport blocks for transmission.

The MAC-scheduler according to the method is activated periodically and for most systems at a fixed MAC-scheduling interval. For GPRS systems such an interval is approximately 20ms and an activating intervals for UMTS-systems is for example nearly every 10ms.

Furthermore, while the PDU-scheduler can read and write in the PDU-queue the MAC-scheduler does only read and implicitly remove the elements in that queue. The reading process is synchronised in such a way that it is assured that any reading access is performed on the queue while the queue is in a valid and consistent state. This could

APPENDIX B

be for example implemented with a semaphore variable. According to the preferred embodiment in a GPRS-System, it is avoided that both schedulers work in an unsynchronised way which also assists to avoid such access conflicts.

Accordingly, the PDU-scheduler operates on the input data e.g. from Layer 3, the Protocol Data Units (PDU). It receives the QoS requirements of each flows and determines the order in which PDU´s should receive service. This service is delivered by lower layer protocol functionality and incorporates mainly two steps. The first is Layer-2 protocol related and includes mainly segmentation of PDU to Transport Blocks (TB) and Automatic Repeat Request (ARQ). The MAC-scheduler serves the PDU's from this list and tries to reflect the order in the list, while also taking power constraints into account. The MAC-scheduler is active at every frame, e.g. on a 20 ms base. The PDU scheduler is operated on all active flows, i.e. with a non-empty PDU-flow-queue. To allow for the linking of the two schedulers the intermediate protocol functionality (depicted as cloud in Figure 1) has to allow for a certain implementation as described below.

According to the method any protocol functionality is defined as operations on data packets. These operations comprise for example an alteration of the packet size, and the content and/or the creation of new packets and are initiated by the arrival from an input source or from another event as for example an interval time out. A method for actively requesting some output can involve the providing of an input and the subsequent waiting for the output or the providing of a clock that triggers

APPENDIX B

internal functionality. However additional asynchronous output may occur as well.

As stated above, the functionality according to the preferred embodiment mainly consists of two processes namely the segmentation and an ARQ. Furthermore additional smaller tasks which are for example the creation of headers or CRC (Cyclic Redundancy Check) creation substantially forming traditional functions on packets that only alter the content of one packet can easily be integrated.

Basing on the aforementioned assumption with regard to Fig. 2 the order of executions of the two schedulers is changed to a scheme of two schedulers forming a poling lower layer and an upper layer which provides the input. Thus, according to the method a linking of the two schedulers is proposed enabling a delivering of merely such an amount of data that the polling lower layer i.e. for example the MAC-scheduler, wants to process. In other words the scheduler on the lower layer provides resources that will have to be filled by the higher layer(s). The principle scheme of an general inter-layer referencing scheme between a delivering upper layer and a polling receiver on a lower layer regardless of the actual implementation of the polling/delivery mechanism is depicted in Fig. 3. The principle scheme for such inter-layer referencing essentially is the following:

a) In an initial step "a" reference to the input is given to the polling receiver;

b) When the receiver polls for new input it puts out a request to the input via the reference. The input might be an actual PDU, the serving-list or a controlling entity that can match QoS requirements

APPENDIX B

and constraints to some PDU-selection. By some means or other, there will be a decision in favour of a flow or PDU;

c) The input in means of a PDU is delivered to some protocol functionality;

d) This will lead to some output. Preferably the implementation is supporting a mechanism that support that this output is well suited to match into the provided resources from the polling lower layer;

e) This output is then taken by the receiver to fill its resources;

f) In case the such encapsulated protocol functionality could generate asynchronous output which has to be expected by the receiver, the receiver additionally has to provide a separate SAP for such asynchronous output.

g) There should also be an SAP offered by the higher layer to allow the encapsulated protocol functionality to signal to the higher layer. Such a signal for example could be the notification of an unsuccessful attempted delivery

It has to be noted that the handling and inputting of a PDU to the protocol functionality is assumed to be one step. Furthermore, the essential reason for distinguishing the steps of outputting, according to point d) of fig. 3, and of delivering to the polling receiver, according to point e) of fig. 3, is based on the fact that the actual polling can happen on a different time scale and thus to allow for various implementations of this timing aspect. There could be for example one delivery performed and a sequence of poll requests of segmented blocks.

APPENDIX B

With the given polling of protocol functionality as exemplary depicted in Fig. 3 the both schedulers are linked together. A general example of such a two-scheduler access on a shared queue is depicted in Fig. 4. It is assumed the output of the PDU-scheduler is the service-list of PDU-packets. The PDU-packets are going to receive service by the MAC-layer. A constraint matching device is determining the serviced PDU's on each MAC-serving interval according to Fig. 4 essentially by the following scheme:

1. The MAC-scheduler states its polling request to a constraint matching device having reading access to the service-list. According to preferred embodiments and for the sake of simplicity this device is assumed to be either part of the MAC-scheduler or part of the PDU-scheduler;
2. The matching device checks the PDU's respectively the PDU-flow properties for finding the first PDU in the service-list that shall receive service and comprises constraints allowing for transmission. Since however the exact mechanism of such an device is highly dependent on the network system used and involves the checking of the system specific constraints, as known by persons skilled in the art, it is not described in detail;
3. Once a decision is made which PDU or PDU-flow shall receive service, it is polled as described with regard to Fig. 3.;
4. The output receives the required protocol functionality and is passed to the MAC-scheduler;
5. As described with regard to Fig. 3, the MAC-scheduler provides an extra input for the asynchronous output involving for example a signalling and ARQ retransmission. A preferred

APPENDIX B

implementation is to grant precedence to input at this SAP.

According to a preferred embodiment, as stated above, the inventive approach is incorporated within a packet switched-system having transceiver units operating on a multi-layer model for protocols. Basing on a GPRS-System, the Logical-Link-Control-Layer (LLC-layer) comprises means for performing the PDU-scheduling and accordingly the MAC-layer comprises means for performing the MAC-scheduling. As known, there is a maximum of 8 timeslots in GPRS-systems for each carrier frequency and several carriers are treated independently of each other as there is no multiplexing between carriers possible. Therefore scheduling is restricted to a single carrier system and other carriers are served by own schedulers. Consequently the intermediate protocol functionality is formed by the Radio-Link-Control-layer (RLC-layer) having as main functions the segmentation and the ARQ.

Based on that exemplary system the segmentation and the ARQ as can be implemented as protocol functionality according to the method as follows:

It is assumed an upper number of maintained flows in the system and each flow can be handled on its own. Furthermore, every PDU in the system has a unique identifier and a currently required segmentation size for each flow can be retrieved at any time and can be stored for each flow. A means which is independent with regard to the data flow is apt to dynamically change this value stored. Moreover each of the PDU's can arrive asynchronously at any time and be stored in a PDU-flow-queue, whereby for each established Transport Block Format (TBF) one PDU-flow-queue is provided.

APPENDIX B

Firstly, it is regarded the timing of the two schedulers prior to a data flow from the input of the PDU to the Physical-layer (PHY-layer).

Thus the specific system goes to a wait-state when a new radio bearer is established and the base transceiver station is initialised.

In the wait-state a MAC-interrupt is made when the PHY-layer requests the RLC-blocks from the MAC-scheduler. This occurs periodically and specifically in GPRS-systems approximately every 20ms. On this interrupt the following sequence could be performed:

1) Delivering of the RLC-blocks to the PHY-layer;
2) Going to the PDU-scheduler's function and running a *new_PDU* service;
3) Going to the *MAC_scheduler;*

With regard to the data flow from the input of a new PDU up to the delivering to the physical layer the following sequences are preferred:

new_PDU:

At some time newly arrived PDU's have to be brought into the PDU-scheduling system. If the system allows for synchronisation mechanisms this can be done concurrently on PDU-arrival in a queue. A more general approach is to do this step always before a new MAC-scheduling cycle is started. As there is a need for synchronisation of flows that were inactive, this step is basically needed to perform this operation. Enabling to look at every flow marked as being inactive can be achieved by an explicit flag or simply by conduction its inactivity from having an empty queue. If there is a PDU for this queue, i.e. if

APPENDIX B

there is a PDU arrived within the last MAC-scheduling interval, the respective queue is marked as being active and the first PDU is taken. For the synchronisation of flows a respective function *schedule_PDU* for this PDU could be called. The procedure is finished when all newly reactivated flows during the last MAC-scheduling interval have been synchronised again.

schedule_PDU:

This is the actual QoS scheduling step. According to some scheduling discipline, for example according to the aforementioned documents of Hui Zhang and J. Cobb et al., the PDU receives a timestamp and is inserted into the service-list with regard to that priority.

MAC_scheduler:

It has to be iterate over all resources involving in GPRS-systems up to 8 timeslots (TS) for one carrier whereby the following steps for each TS have to be performed:

1) Requesting of a read access to the service-list and accessing the first PDU-element;
2) Accessing the PDU-element and testing it for constraints. According to the exemplary embodiment the tested constraints are whether the ARQ-window allows for further transmissions and whether the time slot allocation for this PDU-flow allows for transmission on the currently examined TS;
3) In case of an unsuccessful test, skipping to the next PDU in the service-list and repeating step 2). If however the end of the list is reached, there are no schedulable PDU's available the system has to select a padding RLC-block and to proceed further

with the below step 6);

4) In case of a successful test, performing a request for protocol functionality which would lead according to the exemplary embodiment to a segmentation of one RLC-block with a recent coding scheme, whereby this RLC-block receives its header and its sequence number. If however an ARQ is performed on this flow the RLC-block is given to ARQ to get buffered. The block creation is finished by a CRC-creation and the RLC-block is then returned to the MAC-scheduler;

5) If the segmentation of step 4) has received the last RLC-block of the PDU, i.e. the PDU is just receiving its finishing service the next PDU from this PDU-flow has to be scheduled. If a PDU is available in the flow-queue the first PDU is taken and the function *schedule_PDU* has to be called. If no PDU is available the flow is marked as being inactive;

6) The MAC-scheduler buffers the received block for transmission on the recent TS and the next TS gets scheduled starting at step 1) until the last TS is reached;

7) Going to the above-mentioned wait-state.

It has to be noted that step 5) of the *MAC_scheduler* cycle may also be performed at the end of that *MAC_scheduler* cycle. The difference is that with the proposed solution it is possible for a flow to schedule its new PDU with such high priority to get service within this MAC-interval on the subsequent timeslots. Furthermore, if the system finds itself anywhere else but in the wait-state when the MAC-interrupt arrives this indicates an overload situation.

Fig. 5 and 6 illustrate further examples of a preferred

APPENDIX B

embodiment, wherein Fig. 5 shows a service access from the MAC-scheduler and Fig. 6 shows the case where such access causing the removal of a PDU.

With regard to Fig. 5, the MAC-scheduler performs a read access to the service-list as indicated by reference sign 1A. It starts at the head of the service-list, i.e. accesses the list element which refers to the PDU with the most important priority. For simplicity reasons it is the first list element. In case the first one can not be served due to constraints or should not be served due to some optimising process there will be a search in the serving list for another PDU, which is not described here. Then the content of the service element is read, as indicated by reference sign 2A. The content is itself a reference to a flow-queue, where an associated PDU is stored. This reference is followed, as indicated by reference sign 3A, to access the according flow-queue and its first queuing element, the PDU. According to reference sign 4A, the MAC-scheduler's service is given to this PDU by reading parts of the PDU and delivering this to the intermediate protocol functionality. The output is delivered to the MAC-scheduler as indicated by reference sign 4B.

After the subsequent execution of the regular MAC-service to PDU according to Fig. 4 the MAC-scheduler will ultimately serve the last part of a PDU. This will be noticed by the MAC-scheduler and the following sequence will be initiated with regard to Fig. 6. The element in the serving list is referenced, as indicated by reference sign 1C. Its content is read and then the whole element in the serving list is removed, as indicated by reference sign 2C. The reference from this removed list element is followed to access the flow-queue and its first element,

APPENDIX B

as indicated by reference sign 3C. The former successful completion of service to the PDU means that this PDU is fully processed and therefore is now removed as a whole PDU from the flow-queue. The next element in the flow-queue will reach the head of the queue. For this element the PDU-scheduling discipline algorithm is executed to derive a priority indicator. A new element for the serving-list is created while the position in the service list is determined by the priority indicator, as indicated by reference sign 4C. The content of this list element is the reference to the current queue, i.e. the new PDU at its head.

It is noted that the method is described mainly with regard to a GPRS-system. However, the method also may be applied within other multi-layered systems, for example to scheduling of data flows in the downlink shared channel of a UMTS-system.

EP00310344.7
S. Gault 4

APPENDIX B

1/3

FIG. 1

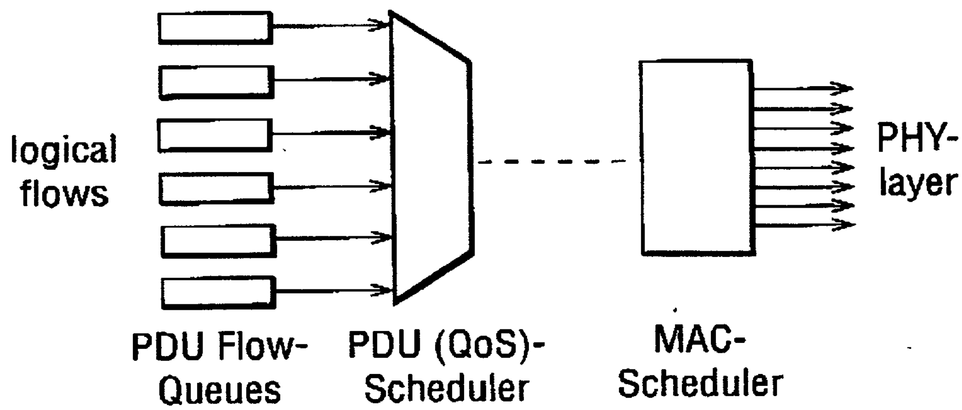
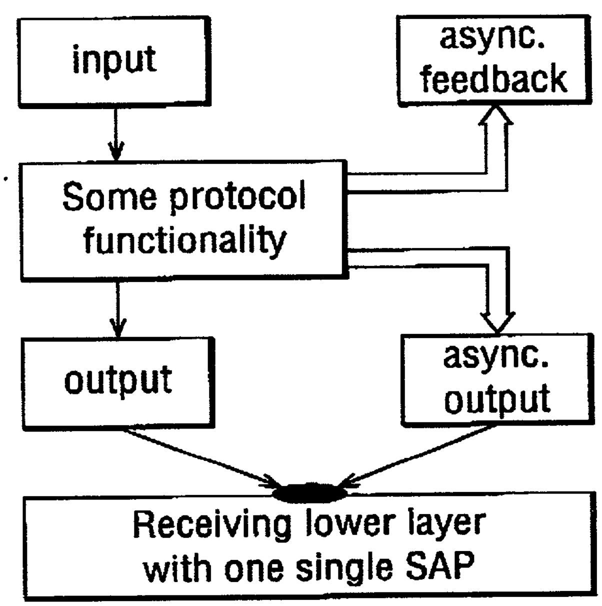


FIG. 2



APPENDIX B

2/3

FIG. 3

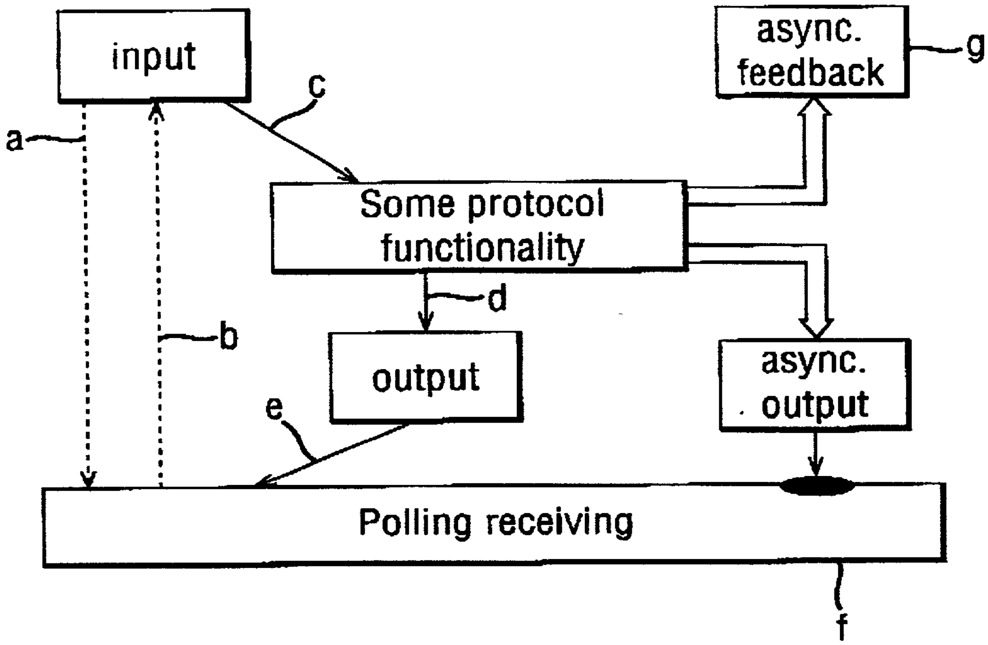
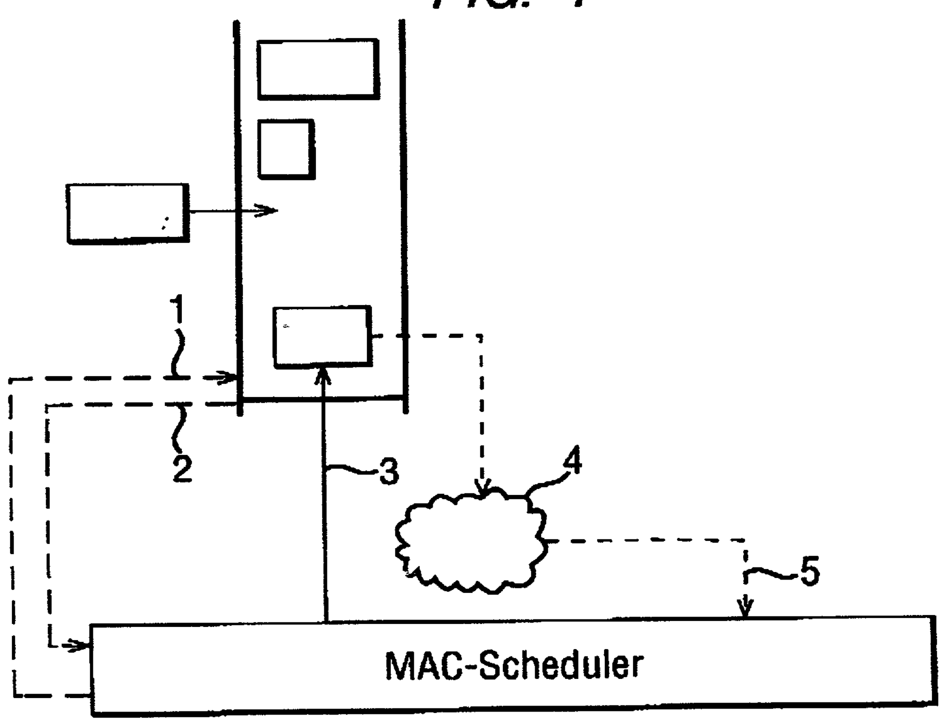


FIG. 4



APPENDIX B

FIG. 5

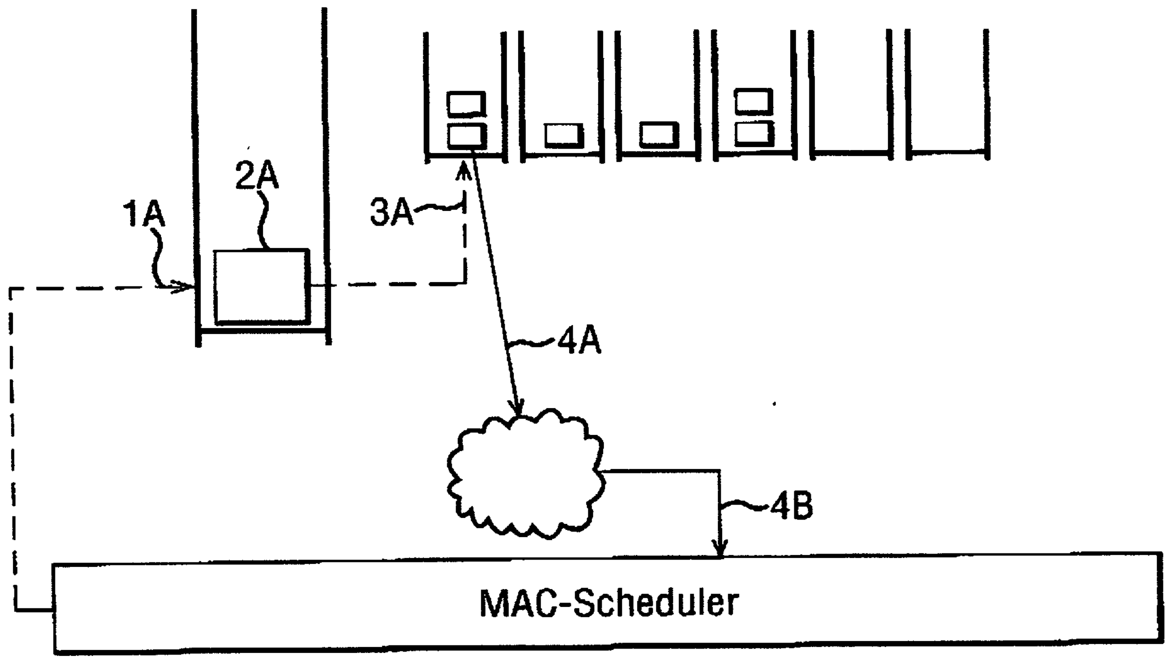
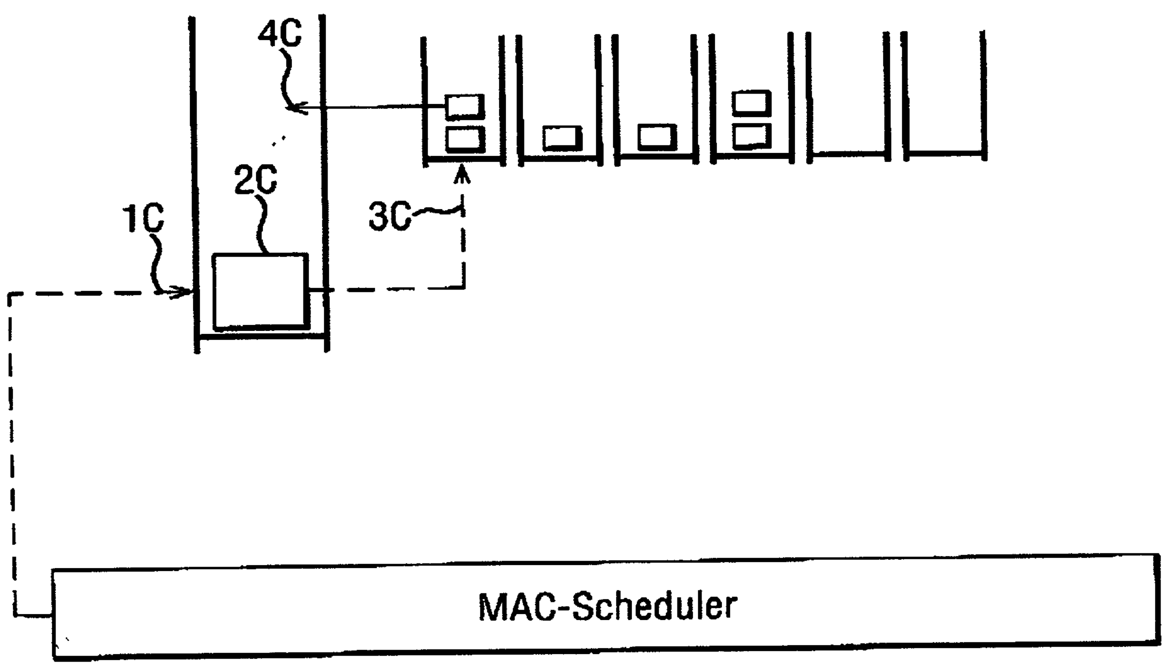


FIG. 6



APPENDIX C

Flow Timestamps

Jorge A. Cobb Amal El Nahas Mohamed G. Gouda

Department of Computer Sciences
The University of Texas at Austin
Austin, Texas 78712-1188

Abstract

We consider the problem of a packet multiplexor that receives packets from multiple input flows, then forwards them to a single output channel. The multiplexor ensures a predefined forwarding rate for each flow. This is achieved as follows. The multiplexor assigns a timestamp to each received packet. When the channel becomes idle, the multiplexor forwards the packet with the smallest timestamp. In this paper, we show that the multiplexor efficiency can be improved without sacrificing packet delay by maintaining one timestamp per flow rather than one timestamp per packet. We show that this approach improves the computational efficiency of two known multiplexing methods, namely, virtual clock and self-clock.

1. Introduction

Virtual circuit computer networks, such as ATM [4], transfer a flow of data packets from a source computer to a destination computer via a virtual circuit. This virtual circuit is established along a network path between the source and destination computers.

Since multiple virtual circuits may share a single output channel of an intermediate computer, a multiplexing algorithm determines the forwarding order of packets along the channel. Some algorithms use packet timestamps to decide the forwarding order [1] [2]. In this paper, we suggest an alternative method that maintains one timestamp per flow, i.e., per virtual circuit, rather than one timestamp per packet. We show that this method improves the computational efficiency without sacrificing the upper bounds on packet delay.

2. Flow Multiplexing

A flow is an infinite sequence of packets that are received by a multiplexor.

The multiplexor receives packets from N distinct flows. A packet is denoted p.(i,k) if it is the kth packet the multiplexor receives from flow i. Note that i is of type 0 . . N-1, and k is a non-negative integer.

We adopt the following notation.

- L.(i,k) is the length in bytes of packet p.(i,k).
- Lmax is an upper bound on all L.(i,k).
- A.(i,k) is the arrival time at the multiplexor of packet p.(i,k).
- E.(i,k) is the exit time from the multiplexor of packet p.(i,k).
- R.i is the rate in bytes/second associated with flow i.

The multiplexor forwards the received packets, one by one, into a single output channel. Thus, whenever the output channel becomes idle, the multiplexor chooses a packet from those it has received, and forwards the packet along the channel.

The goal of the multiplexor is to forward the packets of each flow i at an average rate of at least R.i bytes/second. Since the N flows share the output channel, the following constraint holds, where C is the capacity in bytes/second of the output channel.

$$\sum_{i=0}^{N-1} R.i \leq C$$

3. Multiplexing using Packet Timestamps

A well-known method to multiplex the packets of multiple input flows into a single output channel is to use packet timestamps.

APPENDIX C

When a packet is received, it is assigned an integer, known as the packet's timestamp, then the packet is stored in a queue. This packet queue is shared by all flows, and the packets it contains are ordered by increasing timestamp. When the output channel becomes idle, the packet with the smallest timestamp is removed from the head of the queue and sent over the channel.

Let F.(i,k) be the timestamp of packet p.(i,k). The timestamps of packets are computed such that the channel's bandwidth is shared among the flows according to their rates. In particular, the timestamp F.(i,k) of each p.(i,k) is computed according to the following equation.

$$F.(i,k) = \max(q.(i,k), F(i,k\text{-}1)) + L.(i,k)/R.i \quad (1)$$

In this equation, q.(i,k) is some quantity related to packet q.(i,k). Different methods differ in their choices of q.(i,k). We consider two specific choices for q.(i,k) in Sections 5 and 6 below.

In any case, the algorithm of the multiplexor is as follows.

```
variables
   T.i    :  timestamp of last packet
             received from flow i;
   R.i    :  rate of flow i;
   queue  :  received packets;
   i      :  0..N-1;
   k      :  non-negative integer
begin
    rcv p.(i,k)  →
         T.i := max(q.(i,k), T.i) + L.(i,k)/R.i;
         queue := insert(queue, p.(i,k), T.i)
[]
    idle ∧ queue ≠ empty  →
         p.(i,k) := head(queue);
         queue := tail(queue);
         send p.(i,k)
end
```

This algorithm consists of two actions. In the first action, a packet is received from any flow i, its timestamp is computed, then the packet is inserted into the queue according to the computed timestamp. In the second action, when the output channel becomes idle, the packet with the smallest timestamp is removed from the queue and sent over the channel.

The following theorem shows the relation between the exit time E.(i,k) of packet p.(i,k) and the computed timestamp of p.(i,k).

Theorem 1

If $q.(i,k) \geq A.(i,k)$, then for every i and k,

$$E.(i,k) \leq F.(i,k) + L_{max}/C$$

Proof

Let t be the latest time, $t \leq A.(i,k)$, such that queue was empty, or the packet being sent had a timestamp larger than F.(i,k). Because $q.(i,k) \geq A.(i,k)$, $t \leq A.(i,k) < F.(i,k)$ holds. Also, any packet sent after t and before p.(i,k) is sent must have a timestamp of at most F.(i,k).

Consider any flow j. If at time t flow j has packets in the queue, then their timestamps are greater than F.(i,k), and no packet of j is sent until p.(i,k) is sent. If at time t flow j has no packets in the queue, then the first packet of j arriving after t, say p.(j,l), has a timestamp $F.(j,l) \geq q.(j,l) + L.(j,l)/R.j$. Since $q.(j,l) \geq A.(j,l) \geq t$, $F.(j,l) \geq t + L.(j,l)/R.j$.

Note that, for any l, if $F.(j,l) = F.(j,l\text{-}1) + \Delta$, then $L.(j,l) \leq \Delta * R.j$. Hence, after time t, the packets of flow j that have a timestamp of at most F.(i,k) add to at most $(F.(i,k) - t) * R.j$ bytes. Summing over all flows, the packets arriving at or after t whose timestamp is at most F.(i,k) are at most $(F.(i,k) - t) * C$ bytes. Hence, these packets are sent out in at most $F.(i,k) - t$ seconds. Since there might be a packet being sent at time t with a timestamp greater than F.(i,k), packet p.(i,k) exits no later than

$$t + (F.(i,k) - t) + L_{max}/C$$

End of Theorem 1

4. Multiplexing using Flow Timestamps

From equation (1), it is easy to see that $F.(i,k) > F.(i,k\text{-}1)$, which implies that packets from the same flow are sent in the same order in which they were received. Thus, when a queued packet p.(i,k) has a timestamp smaller than any other queued packet from flow i, it continues to have the smallest timestamp until it is sent, regardless of future packet arrivals from flow i.

APPENDIX C

This observation suggests an alternative scheme, in which only one timestamp per flow is maintained. The packet chosen for transmission is the earliest received packet from the flow with the smallest timestamp.

In this case, the algorithm of the multiplexor is as follows.

```
variables
    U.i     : timestamp of flow i;
    R.i     : rate of flow i;
    queue.i : received packets from flow i;
    i       : 0 . . N-1;
    k       : positive integer
begin
    rcv p.(i,k)    →
        if queue.i = empty     →
            U.i := max(q.(i,k), U.i) + L.(i,k)/R.i
        [] queue.i ≠ empty     →  skip
        fi
        queue.i := append(queue.i, p.(i,k))
[]
    idle ∧ queue ≠ empty      →
        i := min_flow(U);
        p.(i,k) := head(queue.i);
        queue.i := tail(queue.i);
        send p.(i,k);
        if queue.i ≠ empty     →
            p.(i,k) := head(queue.i);
            U.i := U.i + L.(i,k)/R.i
        [] queue.i = empty     →  skip
        fi
end
```

Each flow has a separate first-in-first-out queue of packets, and variable U.i stores the timestamp of flow i.

When packet p.(i,k) is received, if the queue of flow i is not empty, then U.i is not updated. If this queue is empty, then U.i is updated as follows.

$$U.i := \max(q.(i,k), U.i) + L.(i,k)/R.i$$

In the second action, min_flow(U) returns the flow number whose timestamp is the smallest and its queue is non-empty. Thus, the packet to send is taken from the queue of this flow. If more packets of this flow remain, its timestamp is increased by L.(i,k)/R.i, where p.(i,k) is the next packet of the flow.

Flow timestamps have two advantages over packet timestamps.

First, the processing time to send a packet is smaller. This is because the number of queued packets is at least the number of flows with non-empty queues, and hence, finding the minimum flow timestamp is often faster than finding the minimum packet timestamp.

Second, the processing time to receive a packet is also smaller. This is because the flow timestamp is not updated when the flow's queue is not empty. On the other hand, the packet timestamp is always computed upon the packet's arrival, requiring the ordered queue to be updated each time.

To compare flow timestamps with packet timestamps, define G.(i,k) as the timestamp of flow i when packet p.(i,k) becomes the head of the queue of flow i. From the algorithm above, it follows that each G.(i,k) satisfies the following equations. If the queue of flow i is empty at time A.(i,k), then

$$G.(i,k) = \max(q(i,k), G.(i,k-1)) + L.(i,k)/R.i \quad (2)$$

If the queue of flow i is not empty at time A.(i,k), then

$$G.(i,k) = G.(i,k-1) + L.(i,k)/R.i \quad (3)$$

The flow timestamps defined by (2) and (3) are also related to the exit time of the packet as follows.

Theorem 2

If $q.(i,k) \geq A.(i,k)$, then for every i and k,

$$E.(i,k) \leq G.(i,k) + L_{max}/C$$

Proof

Using G.(i,k) instead of F.(i,k), the same argument for the proof of Theorem 1 holds for Theorem 2.

End of Theorem 2

Thus, flow timestamps also provide an upper bound on the exit time of each packet. The relationship between flow timestamps and packet timestamps is given by the following theorem.

Theorem 3

For every i and k,

$$G.(i,k) \leq F.(i,k)$$

Proof Sketch

This is easily shown using induction on the definitions of G.(i,k) and F.(i,k).

End of Theorem 3

From Theorem 3, if a multiplexor uses flow timestamps instead of packet timestamps, its upper bound on packet delay does not increase. Thus, the multiplexor gains processing efficiency without sacrificing packet delay.

We next consider flow timestamps for virtual clock and self-clock multiplexing.

5. Virtual Clock Multiplexing using Flow Timestamps

Virtual clock multiplexing [7] assigns packet timestamps according to equation (1). Its choice of q.(i,k) is q.(i,k) = A.(i,k). Thus, timestamp F.(i,k) of packet p.(i,k) is as follows.

$$F.(i,k) = \max(A.(i,k), F.(i,k\text{-}1)) + L.(i,k)/R.i$$

From Theorem 1, the exit time is as follows.

$$E.(i,k) \leq F.(i,k) + \mathrm{Lmax}/C$$

Using flow timestamps, flow timestamp G.(i,k) is as follows. If the queue of flow i is empty at time A.(i,k), then

$$G.(i,k) = \max(A.(i,k), G.(i,k\text{-}1)) + L.(i,k)/R.i$$

Otherwise,

$$G.(i,k) = G.(i,k\text{-}1) + L.(i,k)/R.i$$

From Theorem 2, the exit time is as follows.

$$E.(i,k) \leq G.(i,k) + \mathrm{Lmax}/C$$

6. Self-Clock Multiplexing using Flow Timestamps

Self-clock multiplexing [3] also assigns packet timestamps according to equation (1). Its choice of q.(i,k) is the timestamp being sent at time A.(i,k). Thus, timestamp F.(i,k) of packet p.(i,k) is as follows,

$$F.(i,k) = \max(F.(j,l), F.(i,k\text{-}1)) + L.(i,k)/R.i$$

where packet p.(j,l) is being sent at time A.(i,k).

Using flow timestamps, the flow timestamp G.(i,k) is as follows. If the queue of flow i is empty at time A.(i,k), then,

$$G.(i,k) = \max(G.(j,l), G.(i,k\text{-}1)) + L.(i,k)/R.i$$

where packet p.(j,l) is being sent at time A.(i,k). Otherwise,

$$G.(i,k) = G.(i,k\text{-}1) + L.(i,k)/R.i$$

It is interesting to note that G.(i,k) = F.(i,k), because of the following. It is easy to show that the timestamps of packets sent by the multiplexor are always in increasing order. Thus, at time A.(i,k), if the queue of flow i is not empty, then q.(i,k) ≤ F.(i,k-1), which implies G.(i,k) = F.(i,k).

Before applying Theorem 1 or 2, we need to show that q.(i,k) ≥ A.(i,k). Due to space limitations, we leave this exercise for the reader. Thus, from Theorem 2, the exit time is as follows.

$$E.(i,k) \leq G.(i,k) + \mathrm{Lmax}/C$$

7. Concluding Remark

We also suspect that another multiplexing method, namely, weighted fair-queuing [2] [5], could benefit from flow timestamps. The packet timestamps in this method satisfy equation (1), and, although computing q.(i,k) is somewhat complex, it appears that q.(i,k) ≥ A.(i,k). If this is the case, Theorem 2 provides a delay bound for the case of flow timestamps.

References

[1] J. Cobb, M. Gouda, "Flow Theory: Verification of Rate-Reservation Protocols", *First IEEE International Conference on Network Protocols*, 1993.

[2] A. Demers, S. Keshav, S. Shenkar, "Analysis and Simulation of a Fair Queueing Algorithm", *Proceedings of the ACM SIGCOM*, 1989.

[3] S. J. Golestani, "A Self-Clocking Fair-Queueing Scheme for Broadband Applications", *Proceedings of the IEEE INFOCOM*, 1994.

[4] F. Halshal, *Data Communications, Computer Networks and Open Systmes*, 3rd Ed., Addison Wesley, 1992.

[5] A. K. J. Parekh, "A Generalized Processor Sharing Approach to Flow Control in Integrated Services Networks", MIT Technical Report No. LIDS-TH-2089.

[6] H. Zhang and Srinivasan Keshav, "Comparison of Rate-Based Service Disciplines", *Proceedings of the ACM SIGCOMM*, 1991.

[7] L. Zhang, "Virtual Clock: A New Traffic Control Algorithm for Packet-Switched Networks", *ACM Trans. on Computer Systems*, 9, 2, May 1991.